



TUGAS AKHIR – TM141585

**REVIEW DAN ANALISA SISTEM MONITORING
PADA WIDYA WAHANA V**

HISYAM FIKRIY
2111100073

Dosen Pembimbing
Dr.Ir. Muhammad Nur Yuniarto

PROGRAM STUDI SARJANA
JURUSAN TEKNIK MESIN
FAKULTAS TEKNIK INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016



TUGAS AKHIR – TM141585

**REVIEW AND ANALISYS OF MONITORING
SYSTEM WIDYA WAHANA V**

HISYAM FIKRIY
2111100073

Faculty Advisor
Dr.Ir. Muhammad Nur Yuniarto

**DEPARTMENT OF MECHANICAL ENGINEERING
FACULTY OH INDUSTRIAL TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2016**

**LEMBAR PENGESAHAN
TUGAS AKHIR (TM141585)**

**REVIEW DAN ANALISA
SISTEM MONITORING WIDYA WAHANA V**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik pada
Program Studi S-1 Jurusan Teknik Mesin
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Disusun Oleh:
HISYAM FIKRIY
NRP. 2111100073

Disetujui oleh Tim Penguji Tugas Akhir:

1. Dr. Ir. M. Nur Yuniarto
NIP. 197506301998621001(Pembimbing)
2. Alief Wikarta, ST, MSc. Eng, PhD
NIP. 198202102006041002(Penguji I)
3. Indra Sidharta, ST, MSc
NIP. 198006192006041004(Penguji II)
4. Prof. Ir. I Nyoman Sutantra, MSc.
PhD(Penguji III)
NIP. 195106051978031002

**SURABAYA
JULI 2016**

REVIEW DAN ANALISA SISTEM MONITORING PADA WIDYA WAHANA V

Nama Mahasiswa : Hisyam Fikriy
NRP : 2111 100 073
Jurusan : Teknik Mesin FTI-ITS
Dosen Pembimbing: Dr. Ir. Muhammad Nur Yuniarto

ABSTRAK

Institut Teknologi Sepuluh Nopember Surabaya (ITS) adalah salah satu perguruan tinggi negeri yang berbasis teknologi terbaik di Indonesia. ITS memiliki peranan cukup penting dalam bidang teknologi yaitu teknologi otomotif di Indonesia. Pada tahun 2013 dan 2015 ini, Institut Teknologi Sepuluh Nopember telah membuktikan bahwa Indonesia dapat membuat mobil dengan tenaga surya dan berajang di perlombaan tingkat Internasional yaitu *World Solar Challenge*. Pada perlombaan tersebut mengharuskan mobil bertenaga untuk menempuh jarak sejauh 3000 km dari Kota Darwin sampai Kota Adelaide tanpa bantuan pengisian daya. Tim Widya Wahana IV hanya mampu menghasilkan jarak tempuh sejauh 784 km dari 3000 km. Salah satu kendala yang terjadi pada Widya Wahana IV adalah tidak terdapat sistem monitoring agar tim saat dapat memanajemen energi tiap harinya sedangkan pada Widya Wahana V sudah terdapat sistem monitoring sehingga hasil jarak yang ditempuh meningkat dari tahun 2013 yaitu 1683 km dari 3000 km. Melalui laporan Tugas Akhir ini, didokumentasikan sistem keseluruhan dari sistem monitoring berupa *hardware* dan *software* yang digunakan pada Widya Wahana V sehingga mendapatkan data akurat secara *real-time* yang terjadi didalam mobil selama perlombaan *World Solar Challenge*.

Dalam tugas akhir ini perancangan sistem monitoring disesuaikan dengan kebutuhan dan spesifikasi pada mobil Widya Wahana V. Sensor DHAB dan *voltage divider* merupakan

hardware yang berfungsi sebagai pengukur arus dan tegangan. Modul Radio dan antena merupakan *hardware* pada sistem telemetri. STM32F3 *Discovery Board* merupakan *hardware* yang berfungsi sebagai *central processing unit*. Pada tugas akhir ini juga akan dirancang *software* khusus yang ditanam pada *hardware* sehingga data dapat diolah. Hasil olahan akan ditampilkan pada display pada mobil Widya Wahana V dan display laptop.

Pada tugas akhir *Review* dan Analisa Sistem Monitoring Pada Widya Wahana V ini didapatkan hasil dokumentasi rancangan dengan 1 *software* utama dan 15 *hardware* utama pada sistem monitoring Widya Wahana V, dokumentasi data *real-time* tiap detik pada sistem monitoring Widya Wahana V selama lomba *World Solar Challenge* 2015, dan terdapat 4 analisa sistem monitoring Widya Wahana V.

Kata Kunci : sistem monitoring, Widya Wahana V, mobil surya

REVIEW AND ANALISYS OF MONITORING SYSTEM WIDYA WAHANA V

Nama Mahasiswa : Hisyam Fikriy
NRP : 2111 100 073
Jurusan : Teknik Mesin FTI-ITS
Dosen Pembimbing: Dr. Ir. Muhammad Nur Yuniarto

ABSTRACT

Sepuluh Nopember Institute of Technology Surabaya (ITS) is one universities based the best technology in indonesia .Its having the role of very important in technology technology called automotive in indonesia .In 2013 and 2015, Sepuluh Nopember Institute of Technology has proved that could make a car with solar cell and perform in a race international level that is World Solar Challenge .In the race requires car power to take the distance of 3000 km from the town of Darwin until Adelaide without the aid of charging power .Team Widya Wahana IV only capable of producing mileage as far as 784 km from 3000 km . One of the obstacles in Widya Wahana IV is there is no monitoring system that the team when can managing energy every day while in Widya Wahana V there had been monitoring system so that the result of the distance traveled increased from year 2013 is 1683 km from 3000 km .Through in final task this report, documented the whole of monitoring system of hardware and software used in widya the v so getting accurate data in real-time occurring in the car during the race world solar challenge.

In final task of this design monitoring system adapted to reflect the needs and specification to car widya the v.Sensors dhab and voltage divider is hardware that serves as a measuring the stream and voltage.Module radio and antennae is hardware on a system telemetry. STM32F3 discovery board is hardware that serves as the central processing unit.On their work end of this will also be designed software special planted in hardware so that data

can be processed. Processing the will be displayed in display to car widya the v and display laptop.

In this final task Review and Analysis of Monitoring System Widya Wahana V showed documentation of design monitoring system Widya Wahana V with 1 major software and 15 major hardware, documentation of real-time data per second monitoring system Widya Wahana V during the race in World Solar Challenge 2015, and there are 4 analysis of monitoring system Widya Wahana V.

Keywords : monitoring system, Widya Wahana V, solar car

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa karena atas anugerha, berkah, dan hidayah-Nya laporan Tugas Akhir yang berjudul “*Review dan Analisa Sistem Monitoring pada Widya Wahana V*” ini dapat diselesaikan.

Dalam kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang sedalam-dalamnya kepada:

1. Bapak Dr.Ir.Muhammad Nur Yuniarto selaku dosen pembimbing yang telah banyak memberikan arahan, bimbingan serta pelajaran selama pembuatan dan penyelesaian mobil Widya Wahana V, selama perlombaan *World Solar Car 2015*, dan juga dalam Tugas Akhir ini.
2. Orang Tua serta seluruh keluarga yang telah banyak memberikan dukungan moral.
3. Seluruh Tim *ITS Solar Car Racing Team 2015* khususnya Jordy dan Seno yang telah banyak membantu dalam pembuatan Tugas Akhir ini.
4. Bengkelerz 2011 yang selalu memberi semangat dan dukungan dalam menyelesaikan tugas akhir ini.
5. Semua pihak yang tidak dapat penulis sebutkan satu-satu.

Penulis menyadari bahwa laporan Tugas Akhir ini masih belum sempurna, baik dari analisis yang penulis lakukan maupun dalam penulisan laporan. Semoga laporan ini dapat memberikan manfaat bagi pembaca pada umumnya dan penulis khususnya.

Surabaya, 9 Juni 2016

“Halaman ini sengaja dikosongkan”

DAFTAR ISI	
HALAMAN JUDUL	i
LEMBAR PENGESAHAN	ii
ABSTRAK	iii
ABSTRACT	v
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penulisan	3
1.5 Manfaat Penulisan	3
BAB II TINJAUAN PUSTAKA	5
2.1 Pendahuluan	5
2.2 Penelitian Terdahulu.....	5
2.2.1 <i>Solar Car Power Monitoring System</i>	5
2.2.2 <i>Wireless Telemetry System for Solar car</i>	6
2.2.2.1 <i>Over-current & Over-voltage Protection Devices</i>	8
2.2.2.2 <i>RPM Measurement</i>	8
2.2.2.3 <i>Current Measurements</i>	9
2.2.2.4 <i>Voltage Measurements</i>	10
2.2.2.5 <i>Temperature</i>	10

2.2.2.6 <i>GPS Receiver</i>	10
2.2.2.7 <i>Wireless Module</i>	10
2.2.2.8 <i>SBC Program</i>	11
2.2.2.9 <i>Remote Laptop Program</i>	12
2.3 <i>Dasar Teori</i>	12
2.3.1 <i>Solar Car</i>	12
2.3.1.1 <i>Sel Surya</i>	13
2.3.1.2 <i>Maximum Power Point Trackers (MPPT)</i>	14
2.3.1.3 <i>Battery Management System</i>	16
2.3.1.4 <i>Baterai</i>	17
2.3.1.5 <i>Motor Controller</i>	18
2.3.1.6 <i>Motor Listrik</i>	18
2.3.2 <i>Sistem Pengukuran Arus Dan Tegangan</i>	19
2.3.2.1 <i>Sensor DHAB</i>	20
2.3.2.2 <i>Voltage Divider</i>	22
2.3.3 <i>Sistem Telemetry</i>	23
2.3.3.1 <i>Modul Radio XBEE</i>	24
2.3.3.2 <i>Modul Radio 3DR</i>	24
2.3.3.3 <i>Antena</i>	25
2.3.3.3.1 <i>Antena Omni</i>	25
2.3.3.3.2 <i>Antena Corner</i>	26
2.3.4 <i>Mikrokontroler</i>	27
2.3.5 <i>Sistem Display</i>	27
2.4 <i>Sistem Komunikasi</i>	28

2.4.1 Komunikasi Serial dan Paralel	28
2.4.2 Jalur CAN.....	29
2.5 Perangkat Pendukung	30
2.5.1 Multimeter.....	30
2.5.2 CAN <i>Reader</i>	31
BAB III METODOLOGI PERCOBAAN	33
3.1 Flowchart Tugas Akhir Sistem Monitoring.....	33
3.2 Review.....	34
3.3 Hasil Review	34
3.4 Analisa Sistem Monitoring.....	34
3.5 Kesimpulan dan Saran.....	34
BAB IV HASIL DAN ANALISA	35
4.1 Sistem Monitoring.....	35
4.2 Flowchart Perancangan Sistem Monitoring	36
4.2.1 Identifikasi Parameter	37
4.2.2 Identifikasi Permasalahan.....	37
4.2.3 Studi Literatur	37
4.2.4 Verifikasi dan Validasi	38
4.2.5 Data Hasil Sistem Monitoring.....	38
4.3 Hasil Review	38
4.3.1 Sistem Monitoring Widya Wahana V	38
4.3.2 Perincian Sistem Monitoring Widya Wahana V	41
4.3.2.1 Sistem Pengambilan Data.....	41
4.3.2.1.1 Pengambilan Data pada Kontroller	41

4.3.2.1.2 Pengambilan Data pada BMS.....	43
4.3.2.1.3 Pengambilan Data pada MPPT.....	44
4.3.2.1.4 Board Pengambilan Data (Joulemeter).....	45
4.3.2.2 Sistem Penampilan Data pada Layar LCD.....	47
4.3.2.3 Sistem Pengiriman dan Penerimaan Data.....	49
4.3.2.3.1 Board Sistem Pengiriman Data (Xbee)	49
4.3.2.3.2 Antena	51
4.3.2.4 Sistem Penampilan Data pada Display Laptop ..	54
4.4 Data Hasil Monitoring.....	58
4.5 Analisa Sistem Monitoring.....	61
BAB V KESIMPULAN DAN SARAN	65
5.1 Kesimpulan.....	65
5.2 Saran.....	66
DAFTAR PUSTAKA	67
LAMPIRAN.....	69
BIODATA PENULIS.....	135

DAFTAR GAMBAR

Gambar 1.1 Widya Wahana V	2
Gambar 2.1 <i>Power system of solar car</i> (Siti Choirun Nisa, 2014)	5
Gambar 2.2 <i>Ideal Design</i> (Team UA Solar Racing Team, 2010)	6
Gambar 2.3 <i>Block diagram of Solar Jackets Telemetry System</i> (Heather Chang dan kawan-kawan, 2010)	7
Gambar 2.4 <i>Current Measurement</i> (Heather Chang dan kawan- kawan, 2010)	9
Gambar 2.5 <i>Update console display of collected data</i> (Heather Chang dan kawan-kawan 2010)	12
Gambar 2.6 Sel Surya (SOLBIAN).....	13
Gambar 2.7 <i>Typical 75W PV module power/voltage/current at standard test condition</i> (Siti Choirun Nisa, 2014).....	15
Gambar 2.8 Contoh Baterai Li-ion (Panasonic)	17
Gambar 2.9 Motor listrik beserta kontroller (MITSUBA)	19
Gambar 2.11 Sensor DHAB (LEM).....	21
Gambar 2.12 <i>Principle of the open loop transducer</i> (LEM)	22
Gambar 2.13 Rangkaian umum <i>voltage divider</i> (Wikipedia).....	22
Gambar 2.14 Modul Radio Xbee (Digi International)	24
Gambar 2.15 Modul radio 3DR (3DR Radio)	24
Gambar 2.16 Antena Omni	26
Gambar 2.17 <i>Corner-reflector antenna</i>	26
Gambar 2.18 <i>Corner-reflector antenna horizontal-plane pattern</i>	27

Gambar 2.19 Topologi CAN (Grangsang Sotyaramadhani 2014)	29
Gambar 2.20 Multimeter	31
Gambar 2.21 CAN <i>Reader</i>	31
Gambar 3.1 Flowchart Tugas Akhir Sistem Monitoring	33
Gambar 4.1 Flowchart Perancangan Sistem Monitoring	36
Gambar 4.2 Perancangan Sistem Monitoring Internal pada Widy Wahana V	38
Gambar 4.3 Perancangan Sistem Monitoring Eksternal pada Mobil Escort	41
Gambar 4.4 Skema pengambilan data kontroller	41
Gambar 4.5 Program Scanning Data Kontroller	42
Gambar 4.6 Skema pengambilan data BMS	43
Gambar 4.7 Program Scanning Data BMS	43
Gambar 4.8 Skema pengambilan data MPPT	44
Gambar 4.9 Program Pengolahan Data MPPT	45
Gambar 4.10 Skema Board Joulemeter	46
Gambar 4.11 Salah Satu Program Joulemeter	47
Gambar 4.12 Skema penampilan data pada LCD	47
Gambar 4.13 Program Display LCD	48
Gambar 4.14 Desain Display LCD	49
Gambar 4.15 Skema pengiriman data	49
Gambar 4.16 Program Xbee	50
Gambar 4.17 Skema Board Xbee	51
Gambar 4.18 Skema Penerimaan data	51

Gambar 4.19 (a) Tampak depan dan (b) Tampak atas antena corner reflector	54
Gambar 4.20 Salah Satu Program pada Sistem Penampilan Data	56
Gambar 4.21 Mendesain Tampilan Menggunakan Software Qt ..	57
Gambar 4.22 Perancangan Display 1 pada Laptop	57
Gambar 4.23 Perancangan Display 2 pada Laptop	58
Gambar 4.24 STM32F3 Discovery Board.....	63
Gambar 4.25 Micro SD	64

“Halaman ini sengaja dikosongkan”

DAFTAR TABEL

Tabel 2.1 Spesifikasi SOLBIAN FLEX SP137.....	14
Tabel 4.1 Dimensi antena Corner Reflector untuk UHF dan VHF	52
Tabel 4.2 Sebagian Data Hasil Monitoring 1 Pada Hari Pertama	59
Tabel 4.3 Sebagian Data Hasil Monitoring 2 Pada Hari Pertama	60
Tabel 4.4 Analisa Sistem Monitoring.....	61

“Halaman ini sengaja dikosongkan”

BAB I

PENDAHULUAN

1.1 Latar Belakang

Institut Teknologi Sepuluh Nopember Surabaya (ITS) adalah salah satu perguruan tinggi negeri yang berbasis teknologi terbaik di Indonesia. ITS memiliki peranan cukup penting dalam bidang teknologi yaitu teknologi otomotif di Indonesia. Pada tahun 2013 dan 2015 ini, Institut Teknologi Sepuluh Nopember telah membuktikan bahwa Indonesia dapat membuat mobil dengan tenaga surya dan berajang di perlombaan tingkat Internasional yaitu *World Solar Challenge* di Australia dengan mobil bertenaga surya yang bernama Sapu Angin Surya atau dapat disebut Widya Wahana.

World Solar Challenge adalah ajang dimana negara-negara di seluruh dunia bersaing dalam manajemen energi pada mobil bertenaga surya. Pada perlombaan tersebut mengharuskan mobil bertenaga untuk menempuh jarak sejauh 3000 km dari Kota Darwin sampai Kota Adelaide tanpa bantuan pengisian daya. Berbagai kendala dan kesulitan dialami Tim Widya Wahana IV sehingga menyebabkan hasil yang kurang maksimal yang hanya menempuh jarak 784 km dari 3000 km dan sisanya dengan bantuan trailer. Salah satu kendala yang terjadi pada Widya Wahana IV adalah tidak terdapat sistem monitoring agar tim saat itu dapat manajemen energi tiap harinya sedangkan pada Widya Wahana V sudah terdapat sistem monitoring sehingga hasil jarak yang ditempuh meningkat dari tahun 2013 yaitu 1683 km dari 3000 km.



Gambar 1.1 Widya Wahana V

Dalam usaha mendapatkan hasil data *real-time* yang baik, maka sistem monitoring yang dirancang diharuskan sesuai dengan kondisi kendaraan tersebut. Berdasarkan hasil penelitian terdahulu yang ditulis oleh *UA Solar Racing Team* (2010) bahwa sistem yang dibuat dan digunakan oleh tim tersebut harus sesuai dengan kondisi kendaraan, pengambilan data secara keseluruhan masih kurang, akurasi yang didapat saat penelitian 95% dari nilai sebenarnya dan data yang dikirim hanya mencapai 1,5 mil. Sedangkan hasil penelitian Heather Chang dan kawan-kawan (2010) sistem yang digunakan untuk mengirim data yaitu menggunakan *wireless telemetry* dan data yang didapat hampir sempurna hanya kurang pada pembacaan arus yang kurang teliti. Melalui laporan Tugas Akhir ini, didokumentasi sistem keseluruhan dari sistem monitoring berupa *hardware* dan *software* yang digunakan pada Widya Wahana V sehingga mendapatkan data akurat secara *real-time* yang terjadi didalam mobil selama perlombaan *World Solar Challenge 2015* dan hasil review dan analisa sistem monitoring pada Widya Wahana V.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah:

1. Bagaimana dokumentasi rancangan *software* dan *hardware* sistem monitoring Widya Wahana V
2. Bagaimana dokumentasi data *real-time* sistem monitoring Widya Wahana V
3. Bagaimana tinjauan dan analisa rancangan sistem monitoring Widya Wahana V

1.3 Batasan Masalah

Dalam penelitian ini permasalahan dibatasi pada:

1. Data *real-time* didapat pada saat perlombaan *World Solar Challenge* 2015

1.4. Tujuan Penulisan

Tujuan dari tugas akhir ini adalah:

1. Mendokumentasikan hasil rancangan *software* dan *hardware* sistem monitoring Widya Wahana V
2. Mendokumentasikan hasil data *real-time* sistem monitoring Widya Wahana V
3. Meninjau dan menganalisa hasil rancangan sistem monitoring Widya Wahana V

1.5. Manfaat Penulisan

Manfaat dari tugas akhir ini adalah:

1. Menjadi acuan saat perlombaan berlangsung
2. Dapat memprediksi target harian dari Widya Wahana V
3. Tim ITS Solar Car Racing Team 2017 mengetahui hasil review dan analisa kekurangan sistem monitoring pada Widya Wahana V

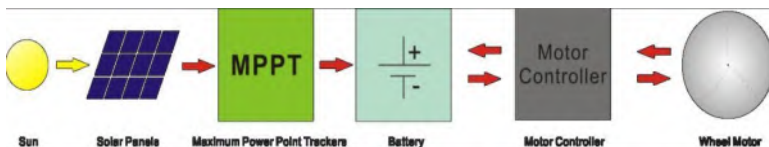
“Halaman ini sengaja dikosongkan”

BAB II

TINJAUAN PUSTAKA

2.1 Pendahuluan

Pada dasarnya prinsip kerja mobil tenaga surya hampir sama dengan prinsip kerja mobil listrik, hanya berbeda beberapa hal. Perbedaan yang paling utama adalah pada mobil listrik pengisian daya baterai menggunakan sumber listrik berupa listrik PLN ataupun semacamnya, sedangkan pada mobil tenaga surya pengisian daya baterai menggunakan sel surya yang dimana sel surya tersebut mengubah energi surya dari matahari menjadi energi listrik. Mobil listrik ataupun mobil tenaga surya terdiri dari dua subsistem dalam sistemnya, motor listrik sebagai penggerak mobil dan platform kendaraan. Komponen utama dalam mobil tenaga surya yaitu sel surya, MPPT (*Maximum Power Point Tracking*), baterai, *controller*, dan motor listrik.



Gambar 2.1 *Power system of solar car* (Siti Choirun Nisa, 2014)

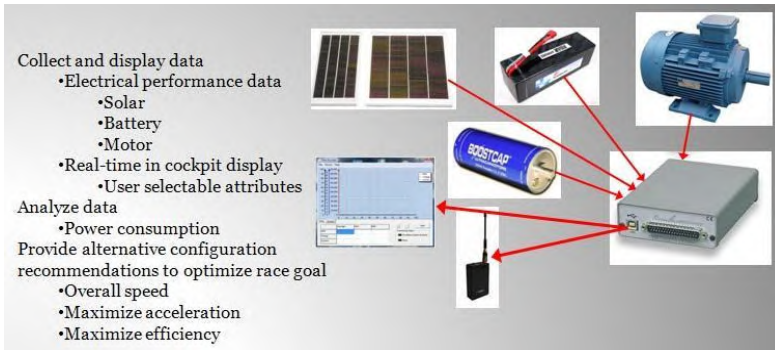
2.2 Penelitian Terdahulu

2.2.1 *Solar Car Power Monitoring System*

Team UA Solar Racing Team (2010) menjelaskan bahwa telah merancang, membangun, dan menguji sistem monitoring untuk mobil *UA Solar Racing Team*. Sistem secara keseluruhan diuji untuk memastikan setiap komponen kompatibel dan bekerja seperti yang dirancang ketika seluruhnya terintegrasi ke dalam kendaraan. Melalui serangkaian analisis dan pengujian dengan data-data yang telah diketahui di setiap bagian dari sistem diverifikasi untuk memenuhi tujuan penggunaan sistem monitoring.

Terdapat dua sistem yang dikembangkan yaitu: sebuah sistem untuk mengevaluasi aspek mekanik *rolling resistance*

melalui *wheel alignment*, dan sistem monitoring listrik untuk mengevaluasi sistem tenaga kendaraan.



Gambar 2.2 Ideal Design (Team UA Solar Racing Team, 2010)

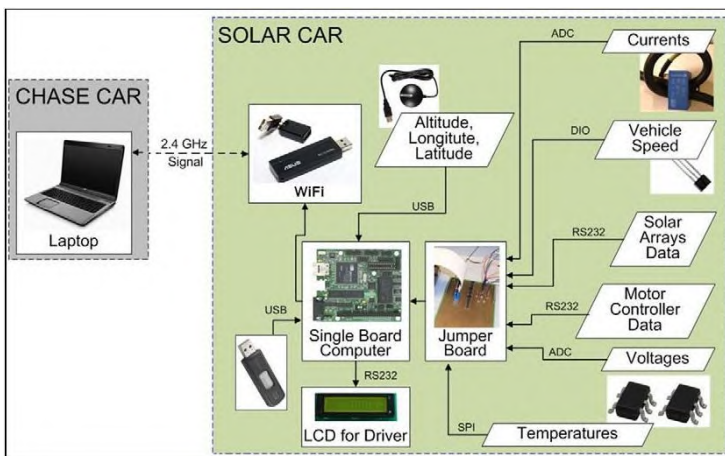
Sistem yang dibuat oleh *Team UA Solar Racing Team* memiliki ketentuan yaitu pertama sistem harus dapat memonitoring kondisi kendaraan dalam akurasi 95% dari nilai yang sebenarnya, merekam dan mengakses melalui sistem telemetri digital secara *real time*. Dan kedua sistem Telemetri memiliki jarak radius berkisar setidaknya 1,5 mil.

2.2.2 Wireless Telemetry System for Solar car

Heather Chang dan kawan-kawan (2010) menyatakan bahwa sistem monitoring adalah subsistem dari desain mobil surya secara keseluruhan, yang meliputi unit untuk pengolahan energi dari sel fotovoltaiik, untuk mekanisme kendaraan, dan untuk kontrol motor. Sedangkan telemetri Sistem adalah, perangkat pemantauan jarak jauh yang mengumpulkan dan mengirimkan data penting penting dari sensor dengan daya rendah dan pengendali dalam mobil surya melalui link nirkabel. Data yang didapatkan ini mencakup tegangan baterai dan arus, tegangan motor controller dan arus, suhu lingkungan indoor dan outdoor, dan kecepatan kendaraan. Sebuah komputer single-board memperoleh dan

memproses sinyal dari sensor dan kontroller dan mengirimkannya ke kendaraan menggunakan standar IEEE 802.11.

Desain kami mampu mengukur sinyal yang disebutkan di atas, kecuali arus. Model yang digunakan saat ini memiliki tegangan output yang terlalu kecil untuk membaca ADC. Dan juga layar LCD untuk pengemudi tidak bisa diprogram karena kendala waktu. Data yang disediakan oleh sistem akan memungkinkan tim untuk dukungan mengembangkan strategi dan memantau kinerja mobil dari luar mobil.



Gambar 2.3 *Block diagram of Solar Jackets Telemetry System*
(Heather Chang dan kawan-kawan, 2010)

Solar Jackets telemetry system terdiri dari tiga komponen penting. Komponen pertama adalah pengumpulan data, yang dilakukan dalam beberapa cara berbeda. Informasi tentang suhu yang dikumpulkan dari sensor analog yang ditempatkan di seluruh mobil surya sementara tegangan diukur menggunakan rangkaian *voltage divider*. Komponen kedua dari sistem telemetri adalah komputer onboard, yang merupakan *single-board computer* (SBC) yang dijalankan Linux. Komputer ini mengumpulkan dan memproses semua data dari komponen-komponen sebelumnya.

Komponen terakhir dari sistem telemetri adalah *wireless* antara komputer onboard dengan laptop portabel. Gambar 2.3 merangkum komponen utama dan *interface* yang digunakan dalam desain kami.

2.2.2.1 Over-current and Over-voltage Protection Devices

sinyal analog keluaran dari sensor dengan ADC (MAX1299) atau SBC, sinyal tersebut dijalankan melalui sepasang menjepit dioda (1N4001) dan *op amp buffer* (LMC6484). Penjepit dioda melindungi ADC dan SBC sebagai masukan dari DIO *lines* dari SBC yang hanya menerima rentang dari 0 sampai 3.3V. Tegangan lonjakan di sirkuit mungkin secara tidak sengaja yang disebabkan oleh ESD (*electrostatic discharge*) dari pengguna atau komponen dengan sirkuit pendek.

CMOS *quad rail-rail op amp* (LMC6484) digunakan sebagai penyangga sebelum memberikan sinyal ke input pin DIO1 atau ADC. Memiliki impedansi *input* yang tinggi dan impedansi *output* yang rendah, buffer memungkinkan komponen yang akan teraliri. *Interfacing output* dari RPM-circuit dengan salah satu DIO1 *line* tanpa op amp, menyebabkan arus lebih tinggi. Akibatnya, output dari rpm-circuit tidak mencapai output tegangan tinggi (2.4V - 3.3V) yang diakui oleh SBC.

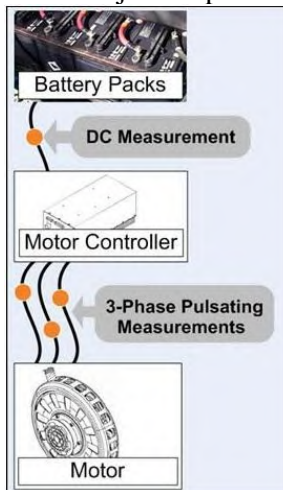
2.2.2.2 RPM Measurement

A1120 adalah *unipolar Hall Effect sensor* yang digunakan untuk pengukuran kecepatan mobil. *Output* rata-rata dari RPM-circuit tanpa magnet adalah 2.8V. Kode SBC menghitung jumlah *Low* dalam rentang waktu tertentu untuk menghitung RPM. Kemudian kecepatan mobil dapat dihitung dengan skala pengukuran RPM dengan mengetahui lingkaran roda. Secara fisik, magnet akan ditempatkan di salah satu bagian depan (*non-motor*) roda sementara sensor hall-efek akan ditempatkan pada mobil surya dekat dengan roda.

2.2.2.3 Current Measurements

Empat set Hass 200-S digunakan untuk mengukur arus di empat lokasi yang berbeda di mobil surya. Satu mengukur arus DC mengalir antara baterai dan motor *controller*. Tiga unit tambahan yang digunakan untuk mengukur arus tiga fase antara kontroler motor dan motor (Gambar 3). Terminal negatif baterai digunakan sebagai dasar untuk semua pengukuran.

Sensor *Hall Effect* sebagai loop terbuka, sedangkan Hass 200-S digunakan untuk mengukur medan magnet. Diukur kemudian dikonversi ke tegangan. Tegangan *output* memasuki salah satu terminal negatif dari op amp (LMC6484), sebelum dimasukkan ke dalam 12-bit ADC (MAX11633) di mana tegangan *output* dikonversi ke nilai biner di antara 0 dan 2^{11} . Setelah itu dimasukkan ke ADC, *Single Board Computer* sebagai kode untuk mengkonversi nilai biner kembali ke tegangan asli yang masuk ADC. Nilai tegangan ini kemudian dikonversikan kembali. Flowchart di bawah ini menunjukkan proses pengukuran.



Gambar 2.4 *Current Measurement* (Heather Chang dan kawan-kawan, 2010)

Metode ini bila digunakan untuk mengukur arus memiliki satu masalah. Tegangan keluaran dari sensor arus sangat kecil sehingga ADC tidak dapat mengenalinya.

2.2.2.4 Voltage Measurements

Tegangan diukur dengan menggunakan *voltage divider*. Gambar di bawah menunjukkan rangkaian *voltage divider* digunakan untuk pengukuran tegangan DC yang keluar dari baterai.

2.2.2.5 Temperature

MAX129 9 adalah sensor suhu yang digunakan dengan *output*-an nilai biner, yang dibaca sebagai integer oleh SBC. SBC diprogram untuk membagi bilangan bulat dengan angka 8, sebagai pengaturan resolusi suhu. Karena output dalam satuan Kelvin, SBC diprogram untuk mengkonversi skala ke Fahrenheit. Sensor suhu dikonfigurasi dengan SBC melalui SPI bus.

2.2.2.6 GPS Receiver

GlobalSat Bu-353 GPS sebagai penerima yang terhubung ke komputer *onboard* melalui koneksi USB. Penerima GPS ini memungkinkan posisi dalam mengumpulkan informasi mengenai bujur, lintang, dan ketinggian mobil surya. Setelah terhubung, penerima menghasilkan data menggunakan protokol standar industri (NMEA 0183). Data ini disimpan dan diteruskan pada link nirkabel melalui kode yang ditulis pada SBC. GPS menerima output pengukuran setiap detik.

2.2.2.7 Wireless Module

Modul nirkabel yang digunakan pada SBC adalah USB adapter Asus Wireless-G. Modul ini memungkinkan SBC untuk bergabung dengan jaringan nirkabel yang juga dapat terhubung dengan laptop. IEEE 802,11g standar nirkabel yang digunakan beroperasi pada frekuensi 2.4G Hz, dan memungkinkan kecepatan transfer data hingga 56 Mbps. Standar ini tersebar luas dan

dibangun untuk hampir setiap laptop sampai sekarang. Dengan menggunakan standar, maka akan sangat mudah untuk memperluas sistem menggunakan router nirkabel standar dan antenna. Spesifik USB adapter Asus menggunakan driver yang didukung oleh produsen SBC (*Technologic Systems*). Semua driver yang diperlukan dimuat melalui *script startup* yang diciptakan untuk SBC. *Script* juga dapat digunakan untuk memberitahu adaptor Asus tentang yang jaringan nirkabel untuk digabungkan.

Dengan menciptakan jaringan nirkabel ini, jaringan pemrograman terhubung socket yang tersedia untuk digunakan pada SBC. Baik koneksi dengan laptop jarak jauh bisa digunakan, atau *datagrams* individu dapat dikirim dari SBC ke port tertentu melalui *port* tertentu. Selain digunakan untuk menyiarkan data yang dikumpulkan melalui USB, jaringan nirkabel dapat berguna untuk debugging.

2.2.2.8 SBC Program

Hanya ada satu program yang diperlukan untuk sistem telemetri untuk memulai pengumpulan dan penampilan data. Program ini ditulis dalam bahasa C, dan berjalan beberapa proses sekaligus dalam satu lingkaran. Program ini mengumpulkan data dari masing-masing sumber dan akhirnya dijadikan satu dengan dipisahkan koma string yang berisi semua data *real-time*. Data ini dibuat setiap detiknya dan ditampilkan ke laptop. Data tersebut juga disimpan ke USB *flash drive* yang dipasang di SBC sepanjang durasi program.

Baik dari ADC dan sensor suhu yang terhubung ke SBC melalui serial perifer (SPI) bus. Meskipun ada yang dibuat dengan fungsi SPI pada SBC, bus ini dibuat menggunakan "bit-banging" dari DIO pada SBC. Banyak keandalan dan kustomisasi masalah yang dihadapi ketika mencoba untuk menggunakan fungsi SPI.

2.2.2.9 Remote Laptop Program

Program *remote laptop* adalah program yang sangat sederhana yang dapat digunakan untuk memantau dan menyimpan data yang ditampilkan oleh SBC. Program ini telah dikembangkan untuk digunakan pada Linux; Namun, program serupa bisa dengan mudah dibuat untuk digunakan di platform lain. Program ini pada dasarnya adalah "*listening*" server yang menunggu data masuk pada port tertentu. Setiap kali data baru dari SBC diterima, data diperbarui pada layar yang ditunjukkan pada Gambar 2.5. Data juga disimpan ke file CSV pada laptop. File CSV ini kemudian bisa dilihat dengan menggunakan program seperti Excel atau Matlab untuk perhitungan dan grafik.

```
File Edit View Terminal Help
V1: 2.32 V Peak
V2: 2.13 V Peak
V3: 2.05 V Peak
VDC: 57.91 V
I1: 0.09
I2: 0.05
ExtT: 1638.3
IntT: 75.7
RPM: 0.0
$GPRMC,180150.189,V,,,,,,,,081110,,,N*49
```

Gambar 2.5 Update console display of collected data (Heather Chang dan kawan-kawan 2010)

2.3 Dasar Teori

2.3.1 Solar Car

Mobil Surya “Solar Car” Mobil tenaga surya atau tenaga matahari, adalah jenis kendaraan listrik yang menggunakan tenaga matahari sebagai sumber utama energinya. Energi matahari ditangkap dengan menggunakan sel surya kemudian diteruskan ke *maximum power point trackers* (MPPT). Setelah masuk MPPT, energi tersebut

dimasukkan ke dalam paket baterai. Energi dari baterai digunakan untuk menggerakkan motor listrik yang dikontrol oleh *motor controller* untuk memutar roda. Dilengkapi dengan alat control pengatur kecepatan maka mobil ini dapat melaju sesuai dengan kecepatan sesuai dengan kecepatan yang dirancang.

2.3.1.1 Sel Surya

Sebuah sel surya, atau sel fotovoltaik, adalah perangkat listrik yang mengubah energi cahaya langsung menjadi listrik oleh efek fotovoltaik, yang merupakan fenomena fisik dan kimia. Sumber dari sel surya bisa berupa sinar matahari ataupun cahaya buatan. Sel surya digunakan sebagai photodetektor (misalnya detektor inframerah), mendeteksi cahaya atau radiasi elektromagnetik lainnya di dekat kisaran terlihat, atau mengukur intensitas cahaya.

Sel photovoltaic (PV) membutuhkan atribut dasar yaitu Penyerapan cahaya, menghasilkan baik pasangan elektron dan pemisahan pembawa muatan dari jenis yang berlawanan.



Gambar 2.6 Sel Surya (SOLBIAN)

Sel surya yang digunakan pada Widya Wahana V adalah sel surya dari produk SOLBIAN dengan tipe FLEX SP137. SOLBIAN FLEX SP137 memiliki spesifikasi seperti pada tabel berikut:

Tabel 2.1 Spesifikasi SOLBIAN FLEX SP137

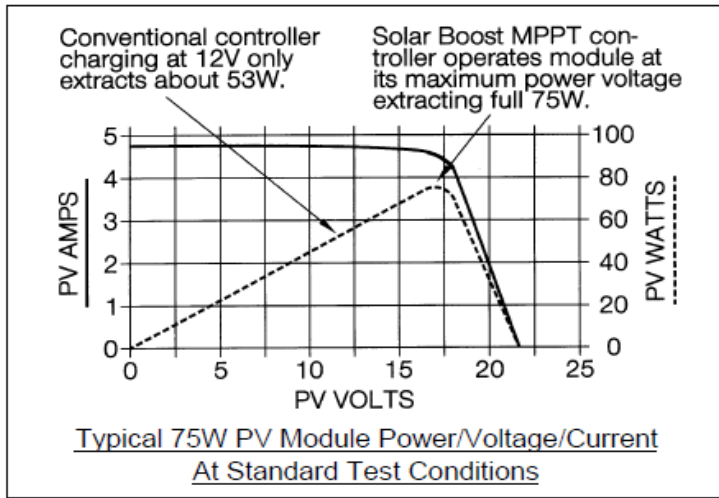
ELECTRICAL CHARACTERISTICS	
Peak Power (+/- 5%) - Pmax	137 W
Rated Voltage - Vmp	24.0 V
Rated Current - Imp	5.7 A
Open Circuit Voltage - Voc	29.1 V
Short circuit Current - Isc	6:00 AM
Temp. coeff. Pmax	-0.38%/°C
Temp. coeff. Voc	-0.27%/°C
Temp. coeff. Isc	0.05%/°C
PHYSICAL CHARACTERISTICS	
Lenght	1490 mm
Width	546 mm
Thickness	2 mm
Weight	2 kg
Num. of cells	44

Efisiensi sel surya dipengaruhi oleh temperatur, jadi sangat penting menjaga suhu sel surya agar tetap pada suhu normal. Oleh sebab itu biasanya harus dilakukan penyemprotan air terhadap sel surya untuk menjaga suhunya. Air memungkinkan untuk menghalangi sinar matahari yang akan masuk ke dalam sel surya, namun kelembapan air itu sendiri jauh lebih penting karena akan meningkatkan 20% *array power* saat cuaca panas.

2.3.1.2 Maximum Power Point Trackers (MPPT)

Maximum Power Point Tracking atau sering disingkat dengan MPPT merupakan sebuah sistem elektronik yang dioperasikan pada sebuah panel *photovoltaic* (PV) sehingga panel PV bisa menghasilkan *power* maksimum. MPPT bukan sebuah sistem *tracking* mekanik yang digunakan untuk mengubah posisi modul terhadap posisi matahari sehingga mendapatkan energi

maksimum matahari, melainkan sebuah sistem elektronik yang bisa menelusuri titik power maksimum yang bisa dikeluarkan oleh sebuah PV.



Gambar 2.7 Typical 75W PV module power/voltage/current at standard test condition (Siti Choirun Nisa, 2014)

Gambar di atas menunjukkan berapa besar perbedaan arus jika sebuah modul PV 75W menggunakan MPPT ataupun tidak. Jika PV panel dihubungkan langsung pada sebuah baterai, maka *power* maksimum yang didapat hanya 53Watt pada tegangan 12Volt. Namun jika menggunakan MPPT, *power* maksimum dapat didapat sebesar 75W pada tegangan 17Volt pada arus maksimum sekitar 4,4Ampere. Sehingga didapatkan arus *charge* baterai sebesar :

$$\frac{V_{panel}}{V_{baterai}} \times I_{modul} = \frac{17V}{12V} \times 4,4A = 6,2A \quad \dots(2.1)$$

Dengan kata lain, *power* maksimum yang didapatkan jika tidak menggunakan MPPT adalah sebesar 70,7%. *Power* maksimum

dapat didapatkan ketika efisiensi 100% dari sistem, namun tidak ada yang 100% untuk efisiensi aktual saat ini. *Power* yang didapatkan akan sedikit lebih rendah karena beberapa *power* yang hilang di kabel, sakring, *circuit breakers*, dan pada *solar boost charge controller*.

2.3.1.3 Battery Management System

Battery Management System (BMS) dalam bahasa yang mudahnya adalah sistem pemantauan baterai, namun dalam arti sesungguhnya adalah sistem yang menjaga pada parameter operasional baterai selama pengisian dan pemakaian seperti tegangan, arus, suhu internal dan ambient baterai. BMS biasanya akan memberikan masukan ke perangkat perlindungan yang akan menghasilkan alarm atau melepas baterai dari beban atau salah satu parameter menjadi keluar dari batas.

Sistem seperti ini bukan hanya mencakup pemantauan dan perlindungan terhadap baterai tetapi juga metode untuk menjaga baterai untuk memberikan power penuh ketika dibutuhkan dan untuk metode memperpanjang hidupnya. Ini mencakup segala sesuatu dari mengendalikan rezim pengisian untuk pemeliharaan yang direncanakan.

Ada tiga tujuan utama umum untuk semua Sistem Manajemen Baterai:

1. Melindungi sel-sel atau baterai dari kerusakan
2. Memperpanjang umur baterai
3. Menjaga baterai dalam keadaan di mana ia dapat memenuhi kebutuhan fungsional dari aplikasi untuk yang telah ditentukan.

BMS menggunakan protokol komunikasi CAN dan *compatible* dengan OBD-II. OBD-II adalah standar protokol komunikasi data untuk otomotif. OBD-II untuk komunikasi CAN distandarkan dalam ISO 15765 CAN (250 kBits/s atau 500 kBits/s).

2.3.1.4 Baterai

Baterai adalah perangkat yang terdiri dari dua atau lebih sel elektrokimia yang mengkonversi energi kimia menjadi energi listrik. Setiap sel memiliki terminal positif atau disebut katoda dan terminal negatif atau anoda. Terminal bertanda positif adalah pada energi potensial listrik lebih tinggi daripada terminal ditandai negatif. Terminal bertanda positif adalah sumber dari elektron ketika terhubung ke sirkuit eksternal yang akan mengalir dan memberikan energi ke perangkat eksternal. Ketika baterai terhubung ke sirkuit eksternal elektrolit bergerak sebagai ion yang memungkinkan reaksi kimia dan akan selesai pada terminal terpisah dan memberikan energi ke sirkuit eksternal. Inilah yang dinamakan gerakan yang ion dalam baterai yang memungkinkan arus mengalir dari baterai untuk melakukan pekerjaan.



Gambar 2.8 Contoh Baterai Li-ion (Panasonic)

Baterai terdapat banyak macamnya, salah satunya adalah baterai Li-ion yang digunakan dalam Widya Wahana V ini. Baterai ion litium (Baterai Li-ion) adalah salah satu anggota keluarga baterai yang dapat diisi ulang. Di dalam baterai ini, lithium ion bergerak dari elektroda negatif ke elektroda positif saat dilepaskan, dan kembali saat diisi ulang. Baterai Li-ion memakai senyawa litium sebagai bahan elektrodanya, berbeda dengan litium metalik yang dipakai di baterai litium non-isi ulang.

Baterai ion litium umumnya dijumpai pada barang-barang elektronik konsumen. Baterai ini merupakan jenis baterai isi ulang

yang paling populer untuk peralatan elektronik portabel, karena memiliki salah satu kepadatan energi terbaik dan mengalami kehabisan isi yang lambat saat tidak digunakan. Selain digunakan pada peralatan elektronik konsumen, LIB juga sering digunakan oleh industri militer, kendaraan listrik, dan dirgantara.

2.3.1.5 Motor Controller

Motor controller adalah perangkat atau kelompok perangkat yang berfungsi untuk mengatur dalam beberapa cara yang telah ditentukan untuk kinerja motor listrik. Sebuah *motor controller* dapat termasuk dalam sarana manual atau otomatis untuk memulai dan menghentikan motor, memilih maju atau rotasi terbalik, memilih dan mengatur kecepatan, mengatur atau membatasi torsi, dan melindungi terhadap overloads dan kesalahan. Adapun beberapa tipe *motor controller*:

2.3.1.6 Motor Listrik

Motor listrik adalah mesin listrik yang mengubah energi listrik menjadi energi mekanik. Kebalikan dari motor listrik yaitu generator yang mengkonversi energi mekanik menjadi energi listrik. Didalam mode otomotif yang seperti biasa, motor listrik beroperasi melalui interaksi antara medan magnet dari motor listrik dan arus digunakan untuk menghasilkan kekuatan atau Bergeraknya motor.

Aplikasi dari motor listrik beragam jenisnya seperti kipas, blower, pompa, peralatan mesin, peralatan rumah tangga, dan disk drive. Motor listrik didukung oleh arus searah (DC) sumber, seperti dari baterai, kendaraan bermotor atau oleh arus bolak-balik (AC) sumber, seperti dari jaringan listrik, inverter atau generator.



Gambar 2.9 Motor listrik beserta kontroller (MITSUBA)

2.3.2 Sistem Pengukuran Arus Dan Tegangan

Pengukuran adalah suatu perbandingan antara suatu besaran dengan besaran lain yang sejenis secara eksperimen dan salah satu besaran dianggap sebagai standar. Dalam pengukuran listrik terjadi juga perbandingan, dalam perbandingan ini digunakan suatu alat Bantu (alat ukur). Alat ukur ini sudah dikalibrasi, sehingga dalam pengukuran listrikpun telah terjadi perbandingan. Sebagai contoh pengukuran tegangan pada jaringan tenaga listrik dalam hal ini tegangan yang akan diukur diperbandingkan dengan penunjukan dari Voltmeter.

Pada pengukuran listrik dapat dibedakan dua hal:

1. Pengukuran besaran listrik, seperti arus (ampere), tegangan (volt), daya listrik (watt), dan lain-lain
2. Pengukuran besaran nonlistrik, seperti suhu, luas cahaya, tekanan, dll.

Dalam melakukan pengukuran, pertama harus ditentukan cara pengukurannya. Cara dan pelaksanaan pengukuran itu dipilih sedemikian rupa sehingga alat ukur yang ada dapat digunakan dan diperoleh hasil dengan ketelitian seperti yang dikehendaki. Juga cara itu harus semudah mungkin, sehingga diperoleh efisiensi

setinggi-tingginya. Jika cara pengukuran dan alatnya sudah ditentukan, penggunaannya harus dengan baik pula. Setiap alat harus diketahui dan diyakini cara kerjanya. Dan harus diketahui pula apakah alat-alat yang akan digunakan dalam keadaan baik dan mempunyai klas ketelitian sesuai dengan keperluannya. Jadi jelas pada pengukuran listrik ada tiga unsur penting yang perlu diperhatikan yaitu:

1. cara pengukuran
2. orang yang melakukan pengukuran
3. alat yang digunakan

Dalam hal ini Widya Wahana V dalam mengetahui arus yang dikeluarkan dalam baterai atau yang digunakan dalam menggerakkan motor menggunakan dua macam sensor yaitu sensor ACS dan sensor LEM DHAB. Sedangkan dalam pengukuran tegangan yang tersisa atau yang masih terdapat di baterai menggunakan *voltage divider*.

2.3.2.1 Sensor DHAB

Sensor ini secara prinsip sama dengan sensor ACS, namun dalam bentuk dan penggunaannya sangat berbeda. Sensor DHAB ini paling cocok untuk DC, AC atau *pulse* dalam pengukuran arus di daya tinggi dan biasa diaplikasikan dalam otomotif dengan tegangan rendah. Sensor Ini berisi isolasi galvanik antara sirkuit primer (daya tinggi) dan sirkuit sekunder (sirkuit elektronik).



Gambar 2.11 Sensor DHAB (LEM)

Prinsip dari sensor ini adalah *loop* transduser terbuka menggunakan efek Hall IC. Magnetik induksi B , berkontribusi terhadap munculnya tegangan Hall yang dihasilkan oleh I_p dan arus primer yang akan diukur. Kontrol I_p saat disuplai oleh sumber arus yaitu baterai atau genset (Gambar 2.12). B sebanding dengan:

$$B (IP) = \text{konstan} (a) \times IP \quad \dots(2.2)$$

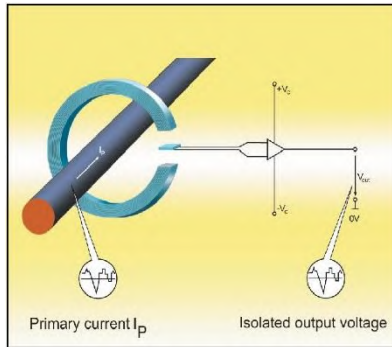
Sehingga tegangan *Hall* dinyatakan dengan:

$$VH = (RH / d) \times I \times \text{konstan} (a) \times IP \quad \dots(2.3)$$

Kecuali untuk IP , semua hal persamaan ini adalah konstan. Oleh karena itu:

$$VH = \text{konstan} (b) \times IP \quad \dots(2.4)$$

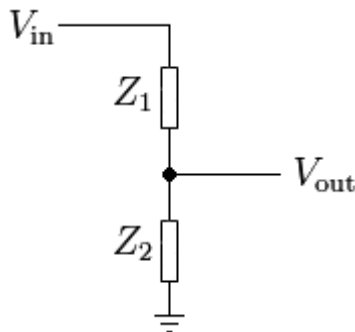
Sinyal pengukuran VH diperkuat untuk memasok tegangan output pengguna atau arus.



Gambar 2.12 Principle of the open loop transducer (LEM)

2.3.2.2 Voltage Divider

Dalam elektronika, *voltage divider* atau juga dikenal sebagai *potensial divider*) adalah sirkuit linear pasif yang menghasilkan tegangan output (V_{out}) yang merupakan sebagian kecil dari tegangan input (V_{in}). *Voltage divider* adalah hasil dari penyebaran tegangan input antara komponen pembagiannya. Contoh sederhana dari *voltage divider* yaitu dua resistor dihubungkan secara seri, dengan tegangan input diterapkan di pasangan resistor dan tegangan output yang muncul dari hubungan antara mereka. *Voltage divider* umumnya digunakan untuk membuat referensi tegangan, atau untuk mengurangi besarnya tegangan sehingga dapat diukur.



Gambar 2.13 Rangkaian umum voltage divider (Wikipedia)

Voltage divider direferensikan ke *ground* dengan menghubungkan dua impedansi listrik dalam seri, seperti yang ditunjukkan pada Gambar 2.13 tegangan input diterapkan di impedansi seri Z_1 dan Z_2 dan output tegangan Z_2 . Z_1 dan Z_2 dapat terdiri dari kombinasi dari unsur-unsur seperti resistor, induktor dan kapasitor. Jika arus dalam kawat output nol maka hubungan antara tegangan input, V_{in} , dan tegangan output, V_{out} , adalah:

$$V_{out} = (Z_2 / (Z_1 + Z_2)) * V_{in} \quad \dots(2.5)$$

Bukti (menggunakan Hukum Ohm):

$$V_{in} = I * (Z_1 + Z_2) \quad \dots(2.6)$$

$$V_{out} = I * Z_2 \quad \dots(2.7)$$

$$I = V_{in} / (Z_1 + Z_2) \quad \dots(2.8)$$

$$V_{out} = V_{in} * (Z_2 / (Z_1 + Z_2)) \quad \dots(2.9)$$

Transfer function (juga dikenal sebagai *voltage ratio*) dari sirkuit ini adalah:

$$H = V_{out} / V_{in} = Z_2 / (Z_1 + Z_2) \quad \dots(2.10)$$

2.3.3 Sistem Telemetri

Sistem telemetri adalah cara pengukuran jarak jauh yang memanfaatkan sarana telekomunikasi dan sistem komputer untuk pengaksesan data. Pada sistem telemetri, semua informasi data diubah ke dalam bentuk informasi listrik dan diolah secara digital. Dengan demikian pada sistem telemeti, semua transduser, sensor, detektor haruslah mempunyai keluaran yang berbentuk besaran elektrik seperti arus (ampere) atau tegangan listrik (volt).

Pemilihan sistem komunikasi yang dipakai disesuaikan dengan kondisi lingkungan dan pertimbangan biaya. Penggunaan satelit adalah pilihan paling mahal, sedangkan penggunaan gelombang radio relatif paling murah. Dengan teknologi yang sudah ada, sistem komunikasi ini dapat mengirimkan data dan ke stasiun pemantau secara transparan (cepat, tidak ada data yang hilang atau berubah). Dalam hal ini Widya Wahana V memakai gelombang radio sebagai akses pengiriman data dan untuk modul radio yang dipakai adalah modul radio Xbee dan modul radio 3DR.

2.3.3.1 Modul Radio XBEE

Xbee adalah nama merek dari Digi International dalam bentuk modul radio yang kompatibel. Modul radio XBee bisa digunakan dengan minimum koneksi dengan *power supply* 3.3 V. Modul ini dikonfigurasi untuk pengiriman jarak jauh tanpa kabel yang dapat diandalkan.



Gambar 2.14 Modul Radio Xbee (Digi International)

2.3.3.2 Modul Radio 3DR

The 3DR Radio adalah cara termudah untuk men-*setup* koneksi telemetry antara APM/Pixhawk dengan perangkat lain semisal laptop. Perangkat ini biasanya memiliki jarak rentang yang lebih baik sejauh 300 m di luar kotak dan dapat diperpanjang untuk beberapa kilometer dengan menggunakan antena.



Gambar 2.15 Modul radio 3DR (3DR Radio)

2.3.3.3 Antena

Perangkat listrik yang mengubah energi listrik menjadi gelombang radio, dan sebaliknya yang disebut antena. Hal ini digunakan pada pemancar radio atau penerima radio. Dalam transmisi, pemancar radio memasok arus listrik berosilasi pada frekuensi radio (yaitu frekuensi tinggi arus bolak-balik (AC)) ke terminal antena dan antena memancarkan energi dari arus gelombang elektromagnetik (gelombang radio). Dalam penerimaan, antena memotong beberapa kekuatan gelombang elektromagnetik untuk menghasilkan tegangan kecil di terminal untuk penerimanya harus diperkuat.

Antena termasuk komponen penting dari semua peralatan yang menggunakan radio. Antena digunakan dalam sistem seperti siaran radio, siaran televisi, radio dua arah, penerima komunikasi, radar, ponsel, dan komunikasi satelit, serta perangkat lain seperti pembuka pintu garasi, mikrofon nirkabel, perangkat berkemampuan Bluetooth, dan jaringan nirkabel komputer.

Biasanya antena terdiri dari susunan konduktor logam (unsur), terhubung elektrik melalui saluran transmisi ke penerima atau pemancar. Elektron dipaksa melalui antena dengan pemancar yang akan membuat medan magnet berosilasi di sekitar elemen antena, sedangkan muatan elektron juga menciptakan medan listrik osilasi sepanjang elemen.

2.3.3.3.1 Antena Omni

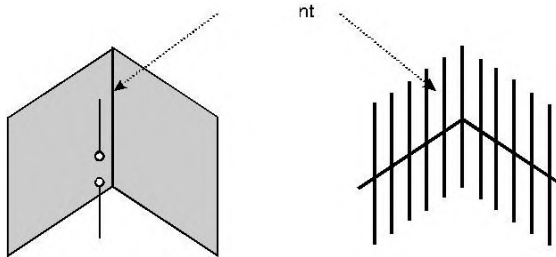
Antena omnidirectional termasuk kelas antena yang memancarkan gelombang radio seragam di semua arah dalam satu perangkat, dengan daya radiasi menurun dengan sudut elevasi di atas atau di bawah perangkat dan jatuh ke titik nol pada sumbu antena. Pola radiasi ini sering digambarkan sebagai bentuk donat. Antena omnidirectional banyak digunakan untuk antena penyiaran radio dan di perangkat *mobile* yang menggunakan radio seperti ponsel, radio FM, *walkie-talkie*, telepon nirkabel, GPS serta untuk BTS.



Gambar 2.16 Antena Omni

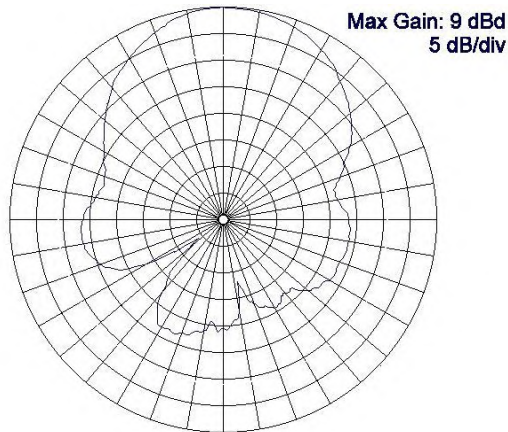
2.3.3.3.2 Antena *Corner*

Antena terdiri dari satu atau lebih elemen dipole di depan reflektor sudut, disebut antena sudut-reflektor, diilustrasikan pada Gambar 2.17.



Gambar 2.17 Corner-reflector antenna

Antena ini memiliki gain cukup tinggi, namun fitur pola yang paling penting adalah bahwa ke depan (balok utama) gain jauh lebih tinggi daripada arah yang berlawanan. Ini disebut rasio *front-to-back* dan dijelaskan dalam pola yang ditunjukkan pada Gambar 2.18.



Gambar 2.18 Corner-reflector antenna horizontal-plane pattern

2.3.4 Mikrokontroller

Mikrokontroller dapat disebut sebagai sebuah komputer kecil yang terletak di satu sirkuit terintegrasi yang terdapat inti prosesor, memori, dan masukan diprogram / output. Memori program dalam bentuk *Ferroelectric* RAM, NOR Flash atau OTP ROM juga sering disertakan *chip*. Mikrokontroller dirancang untuk aplikasi *embedded* berbeda dengan mikroprosesor yang digunakan dalam komputer pribadi. Mikrokontroller yang digunakan dalam suatu produk secara otomatis mengontrol suatu perangkat seperti sistem kontrol mesin mobil, perangkat medis, remote kontrol, dan lain sebagainya.

2.3.5 Sistem Display

Informasi kecepatan, ampere yang digunakan, dan juga informasi voltase baterai sangatlah dibutuhkan dalam mobil surya. Agar *driver* dapat mengetahui informasi tersebut dari sistem display yang dipasang di dalam kendaraan. Dengan demikian driver dapat menentukan kecenderungan untuk tetap melaju kencang atau melaju lambat dengan melihat sistem display yang telah diberikan. Adapun beberapa syarat dalam membuat sistem display yang baik:

1. Sistem display yang tidak mengganggu *driver* dalam mengendarai kendaraan
2. Sistem display yang tidak mengganggu pengelihatannya *driver*
3. Sistem display yang mudah untuk dilihat dengan jelas oleh *driver*

2.3.6 Sistem Operasi Interface

Dalam komputasi, *interface* yaitu batas dimana dua komponen yang terpisah untuk melakukan pertukaran informasi secara sistem komputer. Pertukaran dapat antara perangkat lunak ataupun perangkat keras komputer. Beberapa perangkat keras komputer seperti layar sentuh dapat mengirim dan menerima data melalui sistem operasi *interface*.

Dalam penerimaan data, dibutuhkan software yang dapat menampilkan data secara jelas. Software yang dipakai agar sistem operasi *interface* dapat berjalan yaitu menggunakan *software* QT dimana *software* tersebut dapat berjalan disemua *platform* dan juga menggunakan sistem *opensource* dengan basis UNIX Linux yang telah dimodifikasi.

2.4 Sistem Komunikasi

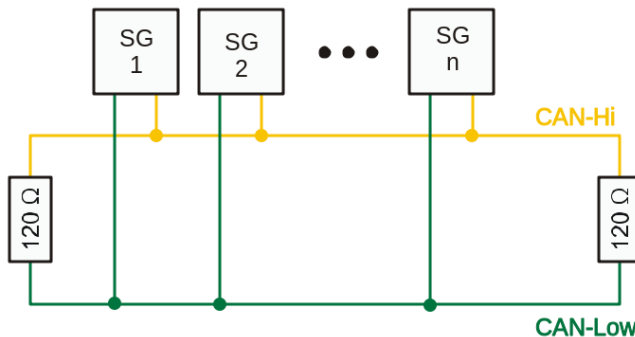
2.4.1 Komunikasi Serial dan Paralel

Komunikasi serial ialah komunikasi yang pengiriman datanya per-bit secara berurutan dan bergantian. Komunikasi ini mempunyai suatu kelebihan yaitu hanya membutuhkan satu jalur dan kabel yang sedikit dibandingkan dengan komunikasi paralel. Pada prinsipnya komunikasi serial merupakan komunikasi dimana pengiriman data dilakukan per bit sehingga lebih lambat dibandingkan komunikasi paralel, atau dengan kata lain komunikasi serial merupakan salah satu metode komunikasi data di mana hanya satu bit data yang dikirimkan melalui seuntai kabel pada suatu waktu tertentu. Pada dasarnya komunikasi serial adalah kasus khusus komunikasi paralel dengan nilai $n = 1$, atau dengan kata lain adalah suatu bentuk komunikasi paralel dengan jumlah

kabel hanya satu dan hanya mengirimkan satu bit data secara simultan. Hal ini dapat disandingkan dengan komunikasi paralel yang sesungguhnya di mana n -bit data dikirimkan bersamaan, dengan nilai umumnya $8 \leq n \leq 128$.

2.4.2 Jalur CAN

Pada pembahasan di atas telah disampaikan bahwa BMS menyediakan protokol komunikasi berupa protokol CAN atau biasa disebut dengan CAN BUS. Oleh sebab itu penulis merasa perlu untuk menjelaskan lebih detail tentang CAN BUS. CAN BUS adalah protokol komunikasi yang umum digunakan dalam dunia Automotive, hampir seluruh kendaraan saat ini menggunakan protokol komunikasi CAN BUS. Protokol komunikasi tersebut digunakan sebagai jalur komunikasi antar komponen di dalam kendaraan. CAN BUS digunakan pada kendaraan karena memiliki topologi yang sangat sederhana. Berikut adalah gambar topologi CAN BUS.



Gambar 2.19 Topologi CAN (Grangsang Sotyaramadhani 2014)

Pada gambar 2.20 dapat diketahui bahwa CAN BUS memiliki dua jalur transmisi data yaitu CAN High (CAN H) dan CAN Low (CAN L) yang menghubungkan titik-titik atau *subsystem*. Berbeda dengan protokol komunikasi yang lain dimana dua jalur dibedakan menjadi jalur penerima dan jalur pengirim.

Pada CAN BUS jalur CAN H dan CAN L difungsikan sebagai keduanya. Sehingga nilai tegangan pada kedua jalur tersebut sama. Perbedaan sinyal pada kedua jalur tersebut hanya nilai tegangannya saja. Pada CAN H tegangan bernilai positif sedangkan pada CAN L nilai tegangan bernilai negatif. Kedua nilai tersebut akan dibandingkan oleh CAN *controller*. Hal tersebut membuat protokol CAN BUS menjadi protokol yang handal karena pengaruh intervensi tegangan tidak akan merusak isi informasi yang dikirimkan. Pada sinyal CAN BUS pesan atau informasi dikemas menjadi satu paket. Paket tersebut diidentifikasi dengan menggunakan ID dengan ukuran 11 bit untuk CAN Standard dan 29 bit untuk CAN Extended. Dengan ukuran 11 bit memungkinkan untuk membuat ID yang berbeda sebanyak 2048 ID. Kemudian setiap ID mampu membawa 8 paket data dengan ukuran 8 bit untuk masing-masing data. Dengan jumlah ID sebanyak 2048 ID dan 8 paket data masing-masing berukuran 8 bit, maka CAN BUS mampu mengirim dan menerima data dalam jumlah yang sangat banyak. Kemampuan tersebut juga didukung dengan kecepatan transfer CAN BUS yang mencapai 1 Megabits/s.

2.5 Perangkat Pendukung

2.5.1 Multimeter

Multimeter atau multitester adalah alat pengukur [listrik](#) yang sering dikenal sebagai VOM (Volt-Ohm meter) yang dapat mengukur tegangan ([voltmeter](#)), hambatan ([ohm-meter](#)), maupun arus ([amperemeter](#)). Ada dua kategori multimeter: multimeter digital atau DMM (*digital multi-meter*)(untuk yang baru dan lebih akurat hasil pengukurannya), dan multimeter analog. Masing-masing kategori dapat mengukur listrik [AC](#), maupun listrik [DC](#).



Gambar 2.20 Multimeter

2.5.2 CAN Reader

CAN *reader* berfungsi sebagai pembaca sinyal komunikasi CAN atau dapat dikatakan lain sebagai alat pendukung pembacaan sinyal CAN. CAN *reader* akan dihubungkan ke salah satu perangkat BMS ataupun controller, dan salah satu sisi lainnya dihubungkan ke laptop. Sinyal CAN dari BMS atau controller akan diteruskan ke laptop agar dapat mengetahui informasi dari BMS seperti tegangan tiap sel baterai, suhu pada baterai, dan lain sebagainya.



Gambar 2.21 CAN Reader

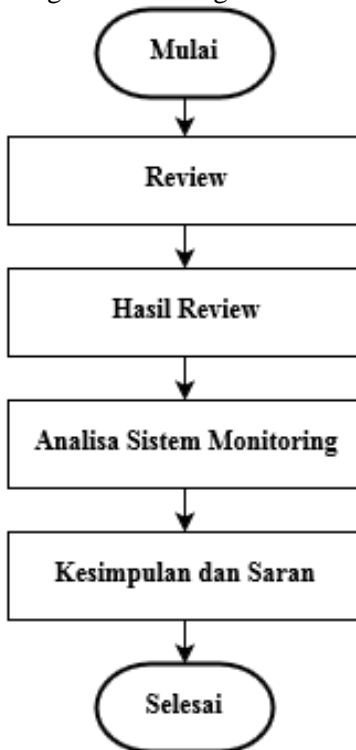
“Halaman ini sengaja dikosongkan”

BAB III

METODOLOGI PENELITIAN

3.1 Flowchart Tugas Akhir Sistem Monitoring

Dalam menyelesaikan tugas akhir ini, disusunlah sejumlah tahapan yang merupakan langkah – langkah dalam mewujudkan tujuan. Tahapan tersebut dinyatakan dalam diagram alir sebagai berikut:



Gambar 3.1 Flowchart Tugas Akhir Sistem Monitoring

Metodologi pelaksanaan tugas akhir ini secara umum dapat dilihat pada flowchart diatas, dimulai dari *review* sistem monitoring pada mobil Widya Wahana V dan pada mobil *escort*.

Kemudian dilanjutkan dengan memaparkan hasil *review* yang telah dilakukan pada tugas akhir ini. Setelah didapat hasil *review* sistem monitoring, baru dilakukan analisa sistem monitoring pada Widya Wahana V dan pada mobil *escort*. Langkah terakhir yaitu memberikan simpulan dan saran mengenai hasil *review* dan analisa sistem monitoring.

3.2 Review

Review sistem monitoring ini dilakukan untuk mengetahui tujuan adanya sistem monitoring, tata cara perancangan sistem monitoring, *software* dan *hardware* yang digunakan pada sistem monitoring, dan hasil yang didapat atau data apa saja yang didapat pada sistem monitoring tersebut.

3.3 Hasil Review

Setelah didapatkan apa saja yang di-*review*, hasil tersebut akan dipaparkan pada tugas akhir ini. Tujuan pemaparan hasil *review* ini agar salah satu orang atau kelompok yang ingin membuat sistem monitoring ini dapat membuat dengan mudah sesuai hasil *review* yang telah dipaparkan.

3.4 Analisa Sistem Monitoring

Analisa sistem monitoring yang dilakukan berupa analisa terhadap kekurangan atau kelemahan sistem monitoring pada saat ini.

3.5 Kesimpulan dan Saran

Tahap akhir dari penulisan tugas akhir ini yaitu penarikan kesimpulan berdasarkan hasil *review* dan analisa sistem monitoring yang telah dilakukan dan pemberian saran agar sistem monitoring selanjutnya lebih baik.

BAB IV

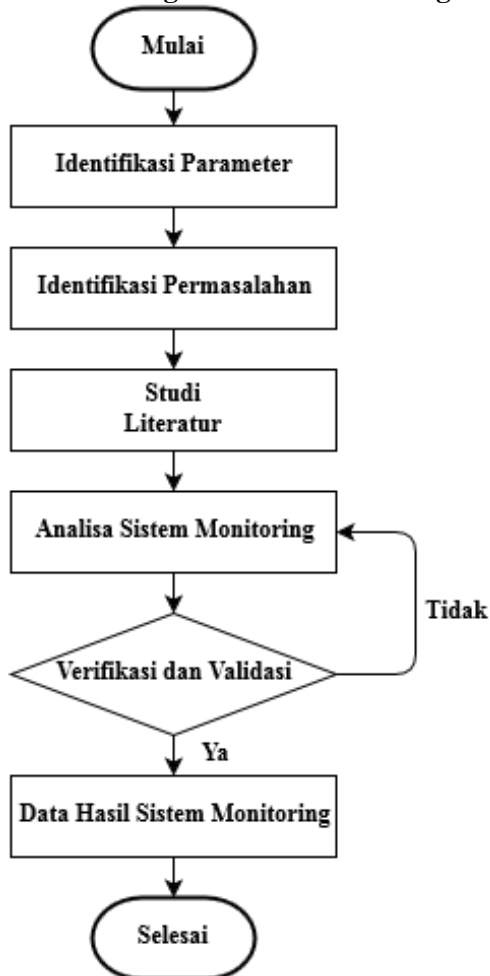
HASIL DAN ANALISA

4.1 Sistem Monitoring

Sistem adalah perangkat unsur yang secara teratur saling berkaitan sehingga membentuk suatu totalitas. Monitoring adalah pemantauan yang dapat dijelaskan sebagai kesadaran tentang apa yang ingin diketahui dan monitoring akan memberikan informasi tentang status dan evaluasi yang diselesaikan dari waktu ke waktu. Dari arti kata itu maka dapat disimpulkan bahwa tujuan sistem monitoring pada umumnya yaitu memastikan suatu proses yang dilakukan sesuai prosedur yang berlaku sehingga proses berjalan sesuai jalur dan dapat mengidentifikasi hasil yang tidak diinginkan pada suatu proses.

Dari tujuan diatas, maka fungsi sistem monitoring pada mobil Widya Wahana V pada saat perlombaan *World Solar Challenge* 2015 yaitu untuk mengetahui konsumsi energi yang terjadi pada mobil Widya Wahana V dan dapat memanajemen energi agar mobil tersebut berjalan sesuai dengan yang direncanakan. Parameter-parameter yang dipilih untuk dimonitoring yaitu kecepatan, arus yang terpakai, arus yang masuk, voltase yang tersisa, dan kondisi baterai. Parameter tersebut dipilih karena memperhitungkan konsumsi energi yang terjadi pada mobil Widya Wahana V dan juga memperhitungkan waktu yang dibutuhkan untuk mencapai tiap *control stop* pada saat perlombaan.

4.2 Flowchart Perancangan Sistem Monitoring



Gambar 4.1 Flowchart Perancangan Sistem Monitoring

Tata cara perancangan sistem monitoring ini secara umum dapat dilihat pada flowchart diatas, dimulai dari identifikasi parameter komponen-komponen yang digunakan. Kemudian

dilanjutkan dengan identifikasi permasalahan apa yang ada pada sistem monitoring, setelah itu baru dilakukan studi literatur mengenai permasalahan-permasalahan yang dihadapi. Setelah mendapatkan studi literatur yang sesuai, maka dilanjutkan menganalisa sistem monitoring. Selanjutnya dilakukan verifikasi dan validasi sistem. Langkah selanjutnya didapatkan data *real-time* sistem monitoring.

4.2.1 Identifikasi Parameter

Identifikasi parameter dilakukan untuk mengetahui karakteristik setiap komponen-komponen yang digunakan. Parameter yang diperlukan yaitu:

- Output MPPT
- Output dan karakteristik BMS
- Output dan karakteristik kontroller
- Karakteristik motor listrik
- Karakteristik sensor DHAB
- Karakteristik modul Xbee
- Karakteristik antena
- Karakteristik display LCD
- Karakteristik mikrokontroller
- *Software* beserta pemrograman yang dibuat

4.2.2 Identifikasi Permasalahan

Identifikasi permasalahan dilakukan setelah mengetahui karakteristik tiap komponen yang digunakan. Dimana indentifikasi permasalahan tersebut untuk mengetahui permasalahan tiap komponen yang akan berhubungan.

4.2.3 Studi Literatur

Studi literatur dilakukan dengan mempelajari berbagai permasalahan yang timbul serta langkah-langkah dan metode yang dapat dilakukan untuk menyelesaikan permasalahan tersebut. Studi literatur dilakukan untuk mendapatkan referensi terkait topik sejenis buku, jurnal, dan berbagai artikel. Studi literatur yang

dilakukan mempelajari beberapa aspek yaitu konsep mobil surya, sistem monitoring internal mobil, dan sistem telemetri.

4.2.4 Verifikasi dan Validasi

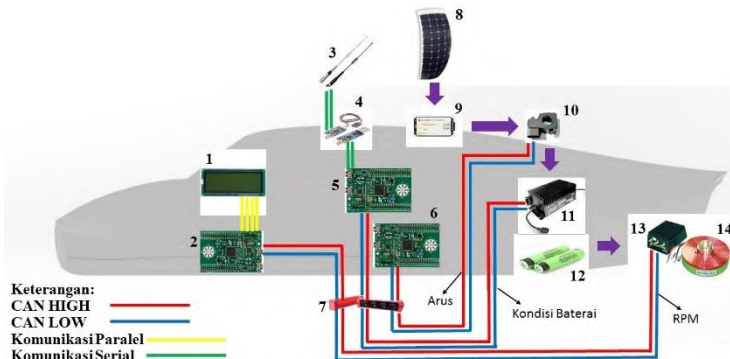
Pada tahap ini analisa sistem monitoring dapat dikatakan hampir selesai, maka diperlukan validasi dan verifikasi. Validasi dan verifikasi dilakukan dengan mensimulasikan sistem monitoring yang dibuat, melihat sistem bekerja sesuai fungsi dan menganalisa *input* ataupun *output* pada *hardware* yang digunakan sesuai parameter yang telah diidentifikasi dengan perangkat pendukung multimeter dan CAN reader. Perangkat pendukung ini hanya untuk pengecekan *output* ataupun *input*, sebagai contoh multimeter untuk mengecek *output* pada MPPT sedangkan CAN reader untuk mengecek *output* dari BMS dan controller.

4.2.5 Data Hasil Sistem Monitoring

Pada tahap terakhir ini didapatkan data secara *real-time* yang telah sesuai dengan keinginan yang membuat sistem monitoring tersebut.

4.3 Hasil Review

4.3.1 Sistem Monitoring Widya Wahana V



Gambar 4.2 Perancangan Sistem Monitoring Internal pada Widya Wahana V

Keterangan:

1. TFT 4" LCD
2. Mikrokontroller 1 (receiver)
3. Antena Omni
4. Modul Xbee
5. Mikrokontroller 2 (transmitter)
6. Mikrokontroller 3 (transmitter)
7. CAN BUS Junction
8. Panel Surya
9. MPPT
10. Sensor DHAB
11. BMS
12. Baterai
13. Kontroller Motor
14. Motor Listrik

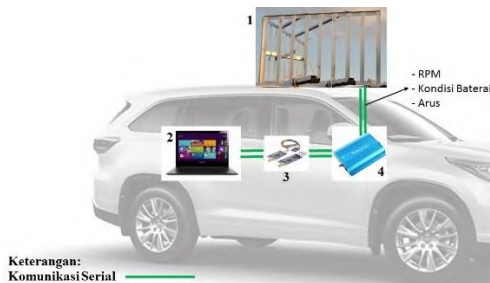
Skema arsitektur pada gambar 4.1 merupakan rancangan yang dapat menjelaskan alur kerja dari sistem monitoring internal pada Widya Wahana V. Sistem penggerak dan sistem penyimpanan energi menyediakan data yang dapat dikumpulkan sebagai informasi performa dari masing-masing komponen pada sistem tersebut. Panel surya mengeluarkan arus listrik menuju MPPT, pada MPPT arus yang mengalir diperkuat/diperbesar semaksimal mungkin. Kemudian arus keluaran MPPT langsung menuju BMS untuk dialirkan langsung ke baterai. Pada *output* MPPT terdapat sensor DHAB yang berguna sebagai sensor arus dimana sensor tersebut digunakan untuk mengetahui *output* dari MPPT yang berupa arus. Sensor DHAB tersebut memiliki karakteristik sebagai berikut yaitu arus yang dapat terbaca berkisar 0-40A dan *output* berkisar 2,5-5V. Dengan begitu *output* dari sensor sebelum masuk mikrokontroller diberi pengondisian signal berupa *voltage divider* agar *output* yang diberikan sebesar 0-3,3V sesuai dengan karakteristik dari STM32 *Discovery Board*. BMS dan kontroller memiliki *output* berupa CAN *High* dan CAN *Low*. Komunikasi

data CAN BUS digunakan pada kendaraan karena memiliki topologi yang sangat sederhana. Berikut merupakan gambaran topologi CAN BUS.

Data yang diambil dari BMS yaitu temperatur setiap sel baterai, voltase setiap sel baterai, total voltase dan ampere yang keluaran maupun yang masuk kedalam baterai, presentase baterai saat *charging*, dan *ampere hour*. Pengambilan data BMS untuk sistem monitoring sesuai dengan tampilan pada *software* BMS. Sedangkan pada kontroller data yang diambil hanya rpm.

Data-data dari MPPT, BMS, dan kontroller akan diolah pada pemrograman mikrokontroller agar data yang ditampilkan sesuai *real-time*. Dalam hal ini STM32 *Discovery Board* atau dapat disebut mikrokontroller berfungsi untuk mengumpulkan data-data tersebut dan mengolahnya. Untuk dapat menjalankan tugas tersebut diperlukan sebuah *firmware* yang ditanamkan pada STM32 *Discovery Board* tersebut. Pada tugas akhir ini nantinya *firmware* akan dibuat dengan menggunakan bahasa C++. *Firmware* tersebut berfungsi untuk mengaktifkan pin CAN-H, CAN-L, *Digital Input*, dan UART (TX-RX) pada STM32 *Discovery Board*. Selain itu *firmware* tersebut juga harus mampu menterjemahkan tipe data CAN dan tipe data digital output menjadi tipe data *serial*.

Setelah pengolahan data pada STM32, salah satu STM32 langsung terhubung dengan display LCD pada mobil Widya Wahana tersebut. Data dikirim melalui komunikasi serial dikarenakan karakteristik dari LCD adalah *input* berupa komunikasi serial, kemudian LCD hanya menampilkan *pack voltage*, *speed*, *rpm*. Pengiriman data ke laptop pada mobil *escourt* melalui antenna Xbee. Xbee memiliki karakteristik yaitu frekuensi kerja pada 900 Mhz, *input* berupa komunikasi serial, dan jangkauan *outdoor* sejauh 50 m, sedangkan untuk *indoor* sejauh 15 m.



Gambar 4.3 Perancangan Sistem Monitoring Eksternal pada Mobil *Escort*

Keterangan :

1. Antena Corner
2. Laptop
3. Modul 3DR
4. *Booster* Antena

Sedangkan pada gambar 4.3 merupakan rancangan yang dapat menjelaskan alur kerja dari sistem monitoring eksternal pada Widya Wahana V. Cara pengambilan data sama dengan sistem internalnya, hanya saja pengiriman data jarak jauh menggunakan antena pada gambar 4.1 dan diterima menggunakan antena pada gambar 4.2 yang dibuat agar dapat menerima data sejauh 700 m. frekuensi disamakan dengan antena pada mobil Widya Wahana sebesar 420 Mhz. Untuk penggunaan antena *booster* yaitu sebagai penguat frekuensi pada antena tersebut.

4.3.2 Perincian Sistem Monitoring Widya Wahana V

4.3.2.1 Sistem Pengambilan Data

4.3.2.1.1 Pengambilan Data pada Kontroller



Gambar 4.4 Skema pengambilan data kontroller

Gambar 4.4 merupakan skema pengambilan data pada kontroller agar data dapat diambil, perlu adanya suatu penghubung antara kontroller dengan *board* joulemeter yaitu berupa kabel beserta konektor untuk meghubungkan kontroller dengan *board*. Data yang diambil menggunakan komunikasi CAN BUS.

```

1  ▶ |** (***|
27
28 /* Define to prevent recursive inclusion -----*/
29 #ifndef __MAIN_H
30 #define __MAIN_H
31
32 /* Includes -----*/
33 #include "ch.h"
34 #include "hal.h"
35 #include "shell.h"
36 #include <string.h>
37 #include "stm32f30x.h"
38 #include "usbcfg.h"
39 #include "chprintf.h"
40
41 /* Exported types -----*/
42 /* Exported constants -----*/
43
44 #define CANx                CAN1
45 #define CAN_CLK             RCC_APB1Periph_CAN1
46 #define CAN_RX_PIN          GPIO_Pin_0
47 #define CAN_TX_PIN          GPIO_Pin_1
48 #define CAN_GPIO_PORT       GPIO_D
49 #define CAN_GPIO_CLK        RCC_AHBPeriph_GPIOD
50 #define CAN_AF_PORT         GPIO_AF_7
51 #define CAN_RX_SOURCE        GPIO_PinSource0
52 #define CAN_TX_SOURCE        GPIO_PinSource1
53
54 //struktur komunikasi serial melalui USB
55
56
57 //definisi CAN Transmit
58 CanTxMsg TxMessage         = {0};
59 CanRxMsg RxMessage         = {0};
60 CanRxMsg RawMessage        = {0};
61
62 CanRxMsg RxMessage12D      = {0};
63 CanRxMsg RxMessage137     = {0};
64
65 //definisi CAN Receive
66 //ID Tegangan
67 CanRxMsg RxMessage33       = {0};
68 CanRxMsg RxMessage44       = {0};
69 CanRxMsg RxMessage55       = {0};
70 CanRxMsg RxMessage66       = {0};

```

Gambar 4.5 Program Scanning Data Kontroller

Sedangkan pada gambar 4.5 yaitu program yang ditujukan untuk mengambil data pada kontroller motor berupa kecepatan motor (rpm).

4.3.2.1.2 Pengambilan Data pada BMS



Gambar 4.6 Skema pengambilan data BMS

Gambar 4.6 merupakan skema pengambilan data pada BMS. Agar data dapat diambil, perlu adanya suatu penghubung antara BMS dengan *board* joulemeter yaitu berupa kabel beserta konektor untuk menghubungkan BMS dengan *board*. Data yang diambil menggunakan komunikasi CAN BUS.

```

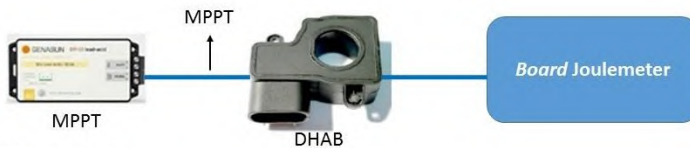
1  // **
27
28 /* Define to prevent recursive inclusion -----*/
29 #ifndef MAIN_H
30 #define MAIN_H
31
32 /* Includes -----*/
33 #include "ch.h"
34 #include "hal.h"
35 #include <string.h>
36 #include "stm32f30x.h"
37 #include "usbcfg.h"
38 #include "chprintf.h"
39
40 /* Exported types -----*/
41 /* Exported constants -----*/
42
43 #define CANx          CAN1
44 #define CAN_CLK       RCC_APB1Periph_CAN1
45 #define CAN_RX_PIN    GPIO_Pin_0
46 #define CAN_TX_PIN    GPIO_Pin_1
47 #define CAN_GPIO_PORT GPIO_D
48 #define CAN_GPIO_CLK  RCC_AHBPeriph_GPIO0
49 #define CAN_AF_PORT   GPIO_AF_7
50 #define CAN_RX_SOURCE GPIO_PinSource0
51 #define CAN_TX_SOURCE GPIO_PinSource1
52
53 //struktur komunikasi serial melalui USB
54 SerialUSBDriver SDU1;
55 BaseSequentialStream* chp =(BaseSequentialStream*) &SDU1;
56 #define usb_lld_connect_bus(usbp)
57 #define usb_lld_disconnect_bus(usbp)
58
59 //definisi CAN Transmit
60 CanTxMsg TxMessage = {0};
61 CanRxMsg RxMessage = {0};
62 CanRxMsg RawMessage = {0};
63
64 //definisi CAN Receive
65 //ID Tegangan
66 CanRxMsg RxMessage33 = {0};
67 CanRxMsg RxMessage44 = {0};
68 CanRxMsg RxMessage55 = {0};
69 CanRxMsg RxMessage66 = {0};

```

Gambar 4.7 Program Scanning Data BMS

Gambar 4.7 yaitu program yang ditujukan untuk mengambil data pada BMS. Data yang diambil yaitu temperatur setiap sel baterai, voltase setiap sel baterai, total voltase dan ampere yang keluaran maupun yang masuk kedalam baterai, presentase baterai saat *charging*, dan *ampere hour*.

4.3.2.1.3 Pengambilan Data pada MPPT



Keterangan:

Arus —————

Gambar 4.8 Skema pengambilan data MPPT

Gambar 4.8 merupakan skema pengambilan data pada MPPT. Agar data dapat diambil, perlu adanya suatu sensor yang terhubung antara MPPT dengan *board joulemeter* yaitu sensor arus (DHAB). Data yang diambil menggunakan voltase keluaran pada sensor. Sensor DHAB tersebut memiliki karekteristik sebagai berikut yaitu arus yang dapat terbaca berkisar 0-40A dan *output* berkisar 2,5-5V. Dengan begitu *output* dari sensor sebelum masuk mikrokontoller diberi pengondisian signal berupa *voltage divider* agar *output* yang diberikan sebesar 0-3,3V sesuai dengan karakteristik dari STM32 *Discovery Board*. Program pengolahan data MPPT pada gambar 4.9 merupakan program perhitungan data yang didapat dari MPPT seperti perhitungan arus pada keluaran MPPT.

```

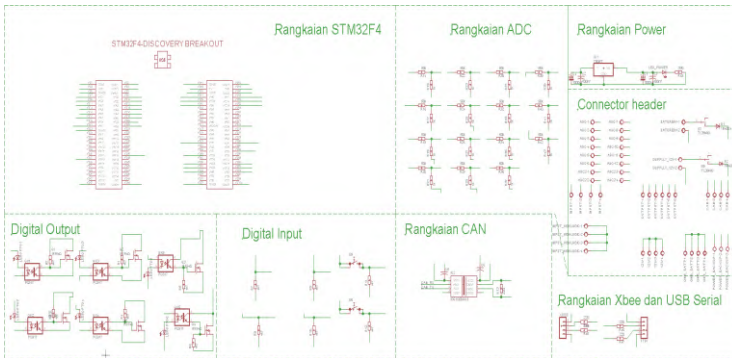
interfacing.cpp  interfacing::perhitungan(): void
544 void interfacing::perhitungan(){
545
546     if(bms_pack.ampereout>6000){
547         (ampereout=bms_pack.ampereout-a);
548         controller_current=((ampereout-panel2_current1)-1);
549     }
550     else{
551         ampereout=bms_pack.ampereout;
552         controller_current=(ampereout-panel2_current1);
553     }
554
555
556     if(speed>6000){
557         (speed=0);
558     }
559
560     if(speed_avg>6000){
561         (speed_avg=0);
562     }
563
564     panel2_current1 = ((panel2_current-3143)/-50);
565     panel1_power = (panel1_current*bms_pack.voltageout);
566     panel2_power = (panel2_current*bms_pack.voltageout);
567     panel3_power = (panel3_current*bms_pack.voltageout);
568     total_power_in =(panel1_power+panel2_power+panel3_power);
569     total_power_out = (controller_power);
570     controller_power = (bms_pack.ampereout*bms_pack.voltageout);
571     controller_voltage = bms_pack.voltageout;
572     panel1_energy += (panel1_power/3600);
573     panel2_energy += (panel2_power/3600);
574     panel3_energy += (panel2_power/3600);
575     controller_energy += (controller_power/3600);
576     current_input = (panel1_current+panel2_current+panel3_current);
577     current_output = (controller_current);
578     energy_input = (panel1_energy+panel2_energy+panel3_energy);
579     energy_output = (controller_energy);
580     energy_terpakai += (controller_energy);
581     energy_terserap += (energy_input);
582     speed_sp_persen=((speed_sp/454)*100);
583     baterai_persen = (((bms_pack.voltageout-75.6)/37.8)*100);
584 }

```

Gambar 4.9 Program Pengolahan Data MPPT

4.3.2.1.4 Board Pengambilan Data (Joulemeter)

Setelah semua alat pengambilan data telah tersedia, maka dibuatlah *board* joulemeter. Dimana *board* tersebut berfungsi sebagai penerima data dari BMS, kontroller, dan MPPT dan juga sebagai pengolah data. Sebelum membuat *circuit board* joulemeter, harus membuat skema elektronik seperti gambar 4.10 agar memudahkan pembuatan *circuit board* joulemeter.



Gambar 4.10 Skema Board Joulemeter

Board joulemeter memiliki dua bagian yaitu bagian atas dan bagian bawah. Bagian atas dan bagian bawah *board* sebagian tersambung satu sama lainnya dikarenakan lintasan (*circuit*) pada board ada beberapa yang bertabrakan. *Board* tersebut tersambung dengan beberapa komponen pada mobil Widya Wahana V yaitu panel surya, BMS, dan kontroller. Jadi dapat dikatakan bahwa fungsi board joulemeter ini sebagai penghitung arus dari panel surya dan penerima data-data dari BMS dan kontroller.

Board joulemeter hanyalah sebagai alat saja, perlu adanya suatu pemrograman didalam *board* tersebut. Pada gambar 4.11 merupakan pemrograman didalam *board* joulemeter yang ditujukan untuk mengolah beberapa data dari BMS agar dapat dibaca berupa ampere yang keluar dan masuk kedalam baterai dan *ampere hour*.

```

1  > /** .....*/
27
28  /* Define to prevent recursive inclusion -----*/
29  #ifndef MAIN_H
30  #define MAIN_H
31
32  /* Includes -----*/
33  #include "ch.h"
34  #include "hal.h"
35  #include <string.h>
36  #include "stm32f30x.h"
37  #include "usbcfg.h"
38  #include "chprintf.h"
39  #include "adc_it.h"
40  #include "my_kirim.h"
41  #include "my_process.h"
42  #include "my_pin.h"
43
44  /* Exported types -----*/
45  /* Exported constants -----*/
46
47  #define CANx                CAN1
48  #define CAN_CLK             RCC_APB1Periph_CAN1
49  #define CAN_RX_PIN          GPIO_Pin_0
50  #define CAN_TX_PIN          GPIO_Pin_1
51  #define CAN_GPIO_PORT       GPIO_D
52  #define CAN_GPIO_CLK        RCC_AHBPeriph_GPIOD
53  #define CAN_AF_PORT         GPIO_AF_7
54  #define CAN_RX_SOURCE        GPIO_PinSource0
55  #define CAN_TX_SOURCE        GPIO_PinSource1
56
57  //struktur komunikasi serial melalui USB
58  SerialUSBDriver SDU1;
59  BaseSequentialStream* chp =(BaseSequentialStream*) &SDU1;
60  #define usb_lld_connect_bus(usbp)
61  #define usb_lld_disconnect_bus(usbp)
62
63  //definisi CAN Transmit
64  CanTxMsg TxMessage         = {0};
65  CanTxMsg TxMessage2        = {0};
66  CanTxMsg TxMessage3        = {0};
67  CanTxMsg TxMessage4        = {0};
68
69  CanRxMsg RawMessage         = {0};

```

Gambar 4.11 Salah Satu Program Joulemeter

4.3.2.2 Sistem Penampilan Data pada Layar LCD



Keterangan:

Komunikasi Paralel —————

Gambar 4.12 Skema penampilan data pada LCD

Gambar 4.12 merupakan skema penampilan data pada LCD. Agar data dapat tertampil dalam LCD, perlu adanya suatu penghubung antara LCD dengan *board* joulemeter yaitu berupa kabel beserta konektor untuk meghubungkan BMS dengan *board*

dan juga sebuah arduino yang terpasang dengan LCD untuk menampilkan display yang telah dirancang. Data yang dikirimkan ke LCD menggunakan komunikasi paralel.

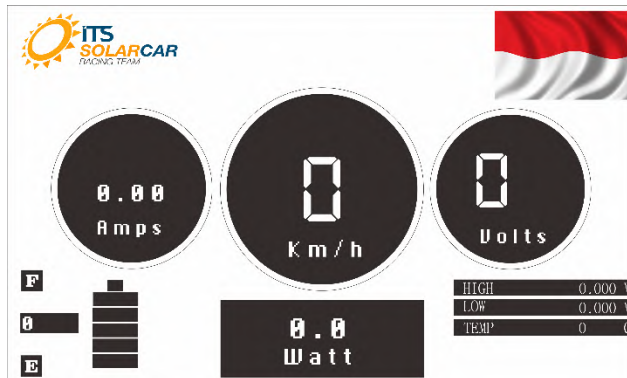
```

1 //memasukkan header file
2 #include "main.h"
3
4 // baca data BMS
5 static WORKING AREA(waBacaBMS, 1024);
6 static msg_t BacaBMS(void *arg) {
7
8     (void)arg;
9     chRegSetThreadName("dataBMS");
10    while (TRUE) {
11
12        //ID 0x1B
13        avgtemp      = RxMessage1B.Data[0];
14        lowvolt1     = RxMessage1B.Data[2];
15        lowvolt2     = RxMessage1B.Data[3];
16        highvolt1    = RxMessage1B.Data[4];
17        highvolt2    = RxMessage1B.Data[5];
18        amphours1    = RxMessage1B.Data[6];
19        amphours2    = RxMessage1B.Data[7];
20
21        //ID 0x6B2
22        packsocraw   = RxMessage6B2.Data[1];
23        packvolt1    = RxMessage6B2.Data[4];
24        packvolt2    = RxMessage6B2.Data[5];
25
26        //ID 0x3B
27        packcurrent1 = RxMessage3B.Data[0];
28        packcurrent2 = RxMessage3B.Data[1];
29        supply12vraw = RxMessage3B.Data[4];
30        packDODraw   = RxMessage3B.Data[5];
31        avgcellvolt1 = RxMessage3B.Data[6];
32        avgcellvolt2 = RxMessage3B.Data[7];
33
34        //ID 0x3CB
35        packDCL      = RxMessage3CB.Data[0];
36        packCCL      = RxMessage3CB.Data[1]; //data Charge Current Limit (CCL)
37        hightemp     = RxMessage3CB.Data[4]; //data Discharge Current Limit (DCL)
38        lowtemp      = RxMessage3CB.Data[5]; //data suhu baterai
39
40        //proses perhitungan
41        avgcellvoltraw = (avgcellvolt1<<8) | (avgcellvolt2);
42        packvoltraw   = (packvolt1<<8) | (packvolt2);
43        lowvolt       = (lowvolt1<<8) | (lowvolt2);
44        highvolt      = (highvolt1<<8) | (highvolt2);

```

Gambar 4.13 Program Display LCD

Dari data-data yang sudah didapatkan, perlu adanya program untuk menampilkan data pada layar LCD pada mobil Widya Wahana V. Dapat terlihat pada gambar 4.13 bahwa program tersebut ditujukan untuk program menampilkan data pada layar LCD dan data yang ditampilkan pada layar LCD yaitu kecepatan, *ampere hour*, total voltase baterai, voltase tertinggi sel baterai, voltase terendah sel baterai dan temperatur baterai. Program tersebut dimasukkan dalam arduino agar data dapat ditampilkan pada LCD sesuai dengan yang didesain.

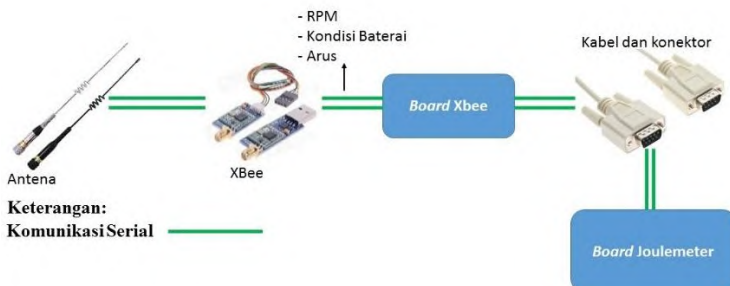


Gambar 4.14 Desain Display LCD

Pada gambar 4.14 merupakan display LCD pada mobil Widya Wahana V. Lingkaran tengah merupakan kecepatan mobil Widya Wahana V, samping kanan merupakan total voltase baterai, samping kiri merupakan *ampere* yang digunakan untuk menggerakkan motor listrik, bagian bawah tengah merupakan daya yang terpakai saat perjalanan, bagian bawah kanan merupakan voltase sel baterai tertinggi terendah dan temperatur baterai, dan pada bagian bawah kiri merupakan persentase baterai yang tersisa.

4.3.2.3 Sistem Pengiriman dan Penerimaan Data

4.3.2.3.1 Board Sistem Pengiriman Data (Xbee)



Gambar 4.15 Skema pengiriman data

Gambar 4.15 merupakan skema pengiriman data menggunakan antenna. Agar data dapat terkirim melalui antenna, perlu adanya suatu penghubung antara antenna dengan *board* joulemeter yaitu berupa kabel beserta konektor untuk meghubungkan *board* joulemeter dengan *board* Xbee dan juga sebuah modul Xbee untuk menghubungkan *board* Xbee dengan antenna. Data yang dikirimkan dari *board* joulemeter hingga antenna menggunakan komunikasi serial sedangkan antar antenna menggunakan gelombang radio.

```

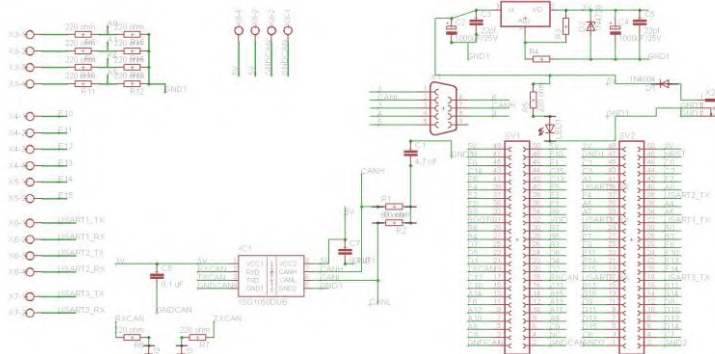
1  //memasukkan header file
2  #include "main.h"
3
4  //baca data board arus
5  static WORKING_AREA(waBacajoule, 1024);
6  static msg_t Bacajoule(void *arg) {
7
8      (void)arg;
9      chRegSetThreadName("dataJoule");
10     while (TRUE) {
11
12         //data tegangan ID 0x111
13         currentjoule1_1 = RxMessage111.Data[0];
14         currentjoule1_2 = RxMessage111.Data[1];
15         currentjoule1 = (currentjoule1_1<<8) | (currentjoule1_2);
16         currentjoule2_1 = RxMessage111.Data[2];
17         currentjoule2_2 = RxMessage111.Data[3];
18         currentjoule2 = (currentjoule2_1<<8) | (currentjoule2_2);
19         currentjoule3_1 = RxMessage111.Data[4];
20         currentjoule3_2 = RxMessage111.Data[5];
21         currentjoule3 = (currentjoule3_1<<8) | (currentjoule3_2);
22         voltjoule1_1 = RxMessage111.Data[6];
23         voltjoule1_2 = RxMessage111.Data[7];
24         voltjoule1 = (voltjoule1_1<<8) | (voltjoule1_2);
25
26         current1_raw = (((float)currentjoule1-2049)/-16.67);
27         current2_raw = (((float)currentjoule2-2062.5)/-17);
28         current3_raw = (((float)currentjoule3-2023)/-16);
29         voltage1_raw = (((float)voltjoule1+15.13)/25.14);
30
31         //data tegangan ID 0x222
32         voltjoule2_1 = RxMessage222.Data[0];
33         voltjoule2_2 = RxMessage222.Data[1];
34         voltjoule2 = (voltjoule2_1<<8) | (voltjoule2_2);
35         voltjoule3_1 = RxMessage222.Data[2];
36         voltjoule3_2 = RxMessage222.Data[3];
37         voltjoule3 = (voltjoule3_1<<8) | (voltjoule3_2);
38         wattjoule1_1 = RxMessage222.Data[4];
39         wattjoule1_2 = RxMessage222.Data[5];
40         wattjoule1 = (wattjoule1_1<<8) | (wattjoule1_2);
41         wattjoule2_1 = RxMessage222.Data[6];
42         wattjoule2_2 = RxMessage222.Data[7];
43         wattjoule2 = (wattjoule2_1<<8) | (wattjoule2_2);
44     }

```

Gambar 4.16 Program Xbee

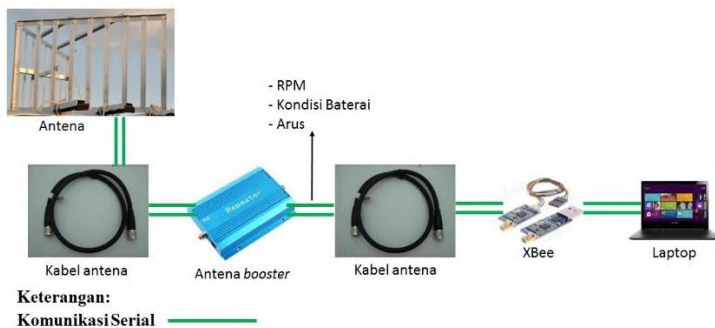
Pada *board* Xbee perlu adanya program dimana program tersebut mengirim data yang diinginkan. Gambar 4.16 ditujukan

untuk program pengiriman data. Data-data yang sudah diolah tersebut dikirim melalui antenna yang terhubung dengan modul Xbee. Data-data yang dikirim akan ditampilkan pada laptop di mobil *escort*. Langkah sebelum membuat *circuit board* Xbee sama dengan joulemeter yaitu harus membuat skema elektronik seperti gambar 4.17 agar memudahkan pembuatan *circuit board* Xbee. *Board* Xbee akan terhubung dengan *board* joulemeter dan modul Xbee. *Board* Xbee ini ditujukan untuk menentukan data-data yang dibutuhkan untuk dikirim melalui modul Xbee.



Gambar 4.17 Skema Board Xbee

4.3.2.3.2 Antena



Gambar 4.18 Skema Penerimaan data

Gambar 4.18 merupakan skema penerimaan data menggunakan antenna. Agar data dapat diterima melalui antenna, perlu adanya suatu penghubung antara antenna dengan laptop yaitu berupa kabel beserta konektor untuk meghubungkan antenna dengan antenna *booster* dan juga menghubungkan antenna dengan modul Xbee. Sedangkan modul Xbee dengan laptop langsung menggunakan USB. Data yang diterima antenna menggunakan gelombang radio sedangkan antenna hingga laptop menggunakan komunikasi serial.

Dalam membuat antenna untuk mengetahui kinerja mobil Widya Wahana V tim instrumentasi mempertimbangkan kebutuhan dengan melihat jangkauan penerimaan ataupun pengiriman data melalui antenna. Karena faktor dalam pembuatan antenna yang kami tekankan yaitu pola radiasi dari antenna, maka kami memilih antenna dengan pola radiasi atau dapat disebut fokus frekuensi yang kedepan, maka antenna *corner reflector* yang cocok dalam hal tersebut. Dan alasan kedua memilih antenna *corner reflector* yaitu dalam perlombaan yang memegang kendali pada instrumentasi adalah mobil *escort* yang berada pada belakang mobil Widya Wahana V. Dengan begitu untuk membuat antenna *corner reflector*, tim melihat pada tabel 4.1 dibawah ini.

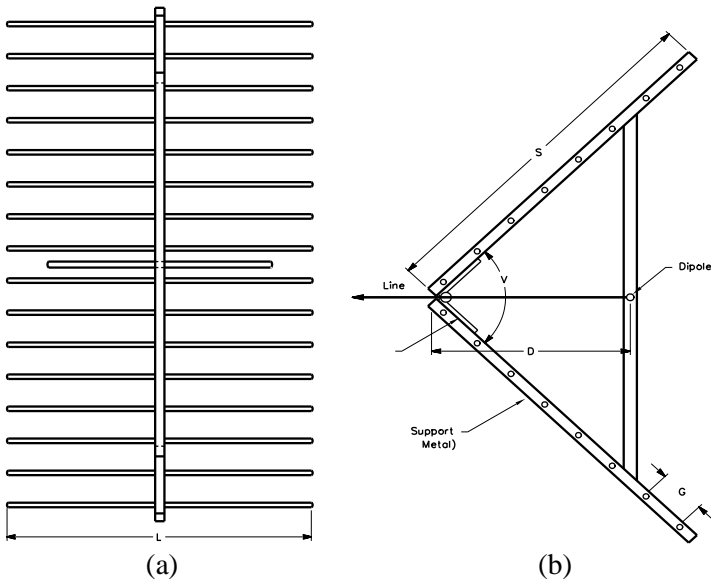
Tabel 4.1 Dimensi antenna Corner Reflector untuk UHF dan VHF

Fre q, MHz	Side Lengt h (S), inch	Dipol e to Verte x (D), inch	Reflect or Length (L), inch	Reflect or Spacing (G), inch	Corn er Angl e, (V°)	Radiation Resistanc e, (Ω)
144	56	27.5	48	7.75	90	70
144	80	40	48	4	90	150
222	42	18	30	5	90	70
222	52	25	30	3	90	150
222	100	25	30	Screen	60	70

420	27	8.75	16.25	Screen	90	70
433	54	13.5	16.25	2.625	60	70
915	20	6.5	25.75	0.65	90	70
915	51	16.75	25.75	Screen	60	65
915	78	25.75	25.75	Screen	45	70
129 6	18	4.5	27.5	0.5	90	70
129 6	48	11.75	27.5	Screen	60	65
129 6	72	18.25	27.5	Screen	45	70
230 4	15.5	2.5	20.5	0.75	90	70
230 4	40	6.75	20.5	Screen	60	65
230 4	61	10.25	20.5	Screen	45	70

Pada tabel 4.1 diambil antena *corner reflector* dengan frekuensi 433. Sebelum memasuki pada pembuatan, pemilihan frekuensi pada frekuensi 433 merupakan frekuensi yang sangat dianjurkan dikarenakan penentuan frekuensi sudah terdapat undang-undang dinegara tempat perlombaan dimana kita dapat menggunakan frekuensi tersebut ataupun tidak, bisa dikatakan legal atau ilegal.

Melihat kembali pada tabel 4.1 dengan memilih frekuensi 433, dimensi yang dipakai yaitu *side length* 54 inch, *dipole to vertex* 13.5 inch, *reflector length* 16.25, *reflector spacing* 2.625 inch, dan dengan sudut 90 derajat. Bentuk dari antena *corner reflector* dapat dilihat seperti pada gambar 4.22 dengan L sebagai *reflector length*, S sebagai *side length*, D sebagai *dipole to vertex*, G sebagai *reflector spacing*, dan V sebagai *corner angle*.



Gambar 4.19 (a) Tampak depan dan (b) Tampak atas antenna corner reflector

4.3.2.4 Sistem Penampilan Data pada Display Laptop

Software untuk sistem ini merupakan software yang berfungsi untuk menampilkan informasi status sistem pada Mobil Widya Wahana V dalam bentuk informasi grafis di laptop *escourt*. Informasi yang diterima merupakan paket data serial yang dikirimkan oleh STM32F3 sesuai sinyal *request* ke STM32F3. Software tersebut ditulis dengan menggunakan bahasa C++ dan QML dengan bantuan software Qt. Pada sistem penampilan data ini, terdapat 8 program yang dibutuhkan agar sistem ini berjalan sesuai dengan yang diinginkan. 8 program tersebut meliputi sebagai berikut:

1. Program untuk layar utama pada program
2. Program pemanggilan data
3. Program pembuatan grafik
4. Program konfigurasi *device*

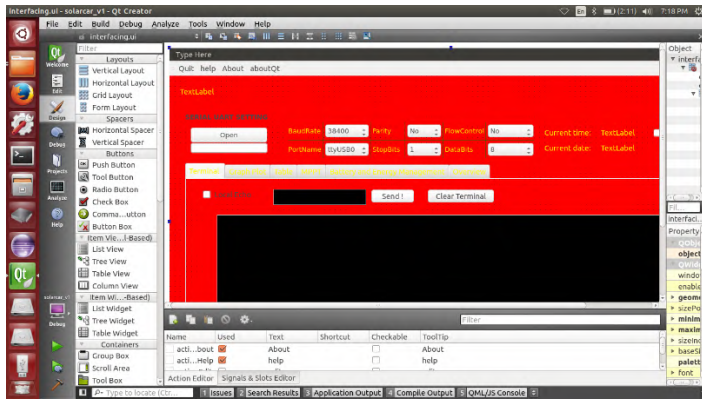
5. Program menampilkan data pada tabel
6. Program pengaturan penyimpanan data
7. Program penampilan data pada display laptop
8. Program penggabungan

Program untuk layar utama merupakan sebuah perawalan dalam menulis sebuah program dan dapat dikatakan sebagai fungsi utama. Fungsi utama tersebut harus ada pada *sourcecode* program sebelum penulisan program lainnya. Program pemanggilan data merupakan program yang berfungsi sebagai memanggil data dengan tipe data yang memiliki angka dibelakang koma, misal kan data *delta cell voltage* sebesar 0,3 volt. Dari program tersebut dapat dibatasi angka dibelakang koma yang diinginkan.

Program pembuatan grafik ditujukan agar dapat melihat grafik konsumsi energi secara *real-time*. Program dari hasil pengolahan data dimasukan pada program pembuat grafik, namun hasil pengolahan data yang dimasukan pada program pembuat grafik ini hanya hasil perhitungan konsumsi energi dan grafik tersebut dapat langsung dilihat pada display laptop.

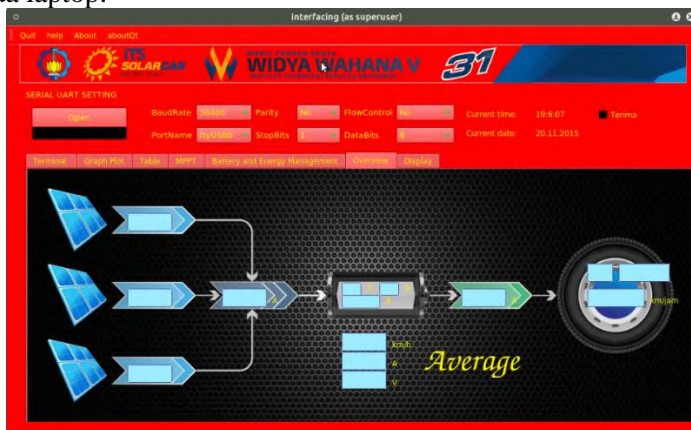
Program konfigurasi device ditujukan agar *device* pendukung seperti modul Xbee yang terhubung dengan laptop dengan data yang dikirim dapat terbaca pada display laptop atau dapat dikatakan sebagai penentu *port* USB laptop yang masih kosong atau belum terpakai. Program penampilan data pada tabel merupakan penampilan data *real-time* yang tersedia pada display laptop yang tersusun pada tabel. Data yang tampil pada tabel ini yaitu kecepatan, posisi *throttle*, rpm, *pack voltage*, arus, *pack current*, *charge current limit*, *state of charge*, *ampere hour*, 12v *supply*, *batt highest cell*, *batt lowest cell*, *batt highest temp*, *batt, lowest temp*, *batt average cell*, dan *batt delta voltage*.

Program pengaturan penyimpanan data merupakan program yang mengatur penyimpan data pada file dengan format ".txt" dan mengatur isi dari file tersebut. Isi dari file tersebut diatur dengan memisahkan per-data yang diambil dengan tanda koma, agar data dapat disusun ulang dengan mudah pada MS Excel. Program penampilan data pada display merupakan program dimana data apa saja yang ditampilkan dan menempatkan data yang ditampilkan sesuai rancangan display pada laptop. Dan terakhir pada pemrograman ini yaitu program penggabungan. Penggabungan yang dimaksud yaitu menyatukan file-file pada file program agar dapat dijalankan pada laptop dengan baik.



Gambar 4.21 Mendesain Tampilan Menggunakan Software Qt

Dalam *software* Qt tersebut kita dapat mendesain display yang diinginkan seperti pada gambar 4.21, namun desain yang dibuat sesuai dengan apa yang dibutuhkan. Sehingga pada Gambar 4.22 hingga gambar 4.23 merupakan hasil perancangan display pada laptop.



Gambar 4.22 Perancangan Display 1 pada Laptop

Pada gambar 4.22 menampilkan display secara keseluruhan, dari rata-rata ampere yang dihasilkan panel surya,

voltase dan ampere dari baterai yang tersisa, ampere yang digunakan atau yang dikeluarkan dari baterai untuk menggerakkan motor, kecepatan mobil surya, dan rata-rata dari kecepatan, voltase, dan ampere selama perlombaan berlangsung.



Gambar 4.23 Perancangan Display 2 pada Laptop

Gambar 4.23 menampilkan display mengenai konsumsi energi dan kondisi baterai, dari temperatur setiap sel baterai, voltase setiap sel baterai, total voltase dan ampere yang keluaran maupun yang masuk kedalam baterai, presentase baterai saat *charging*, dan *ampere hour*.

4.4 Data Hasil Monitoring

Berikut merupakan hasil data monitoring yang tertampil pada display laptop pada mobil *escourt* yang tersimpan pula pada laptop tersebut.

Keterangan Tabel 4.2 dan 4.3:

1. $S = \text{speed (km/h)}$
2. $TP = \text{throttle position (\%)}$
3. $SP = \text{set point rpm}$
4. $PV = \text{pack voltage (V)}$
5. $P = \text{Panel (A)}$

6. PC = *pack current* (A)
7. CCL = *Charge Current Limit*
8. DCL = *Discharge Current Limit*
9. SOC = *State of Charge*
10. AH = *ampere hour* (Ah)
11. 12v = 12 v *supply* (v)
12. BHT = *batt highest temp* (°C)
13. BLT = *batt lowest temp* (°C)
14. BAT = *batt average temp* (°C)
15. BHC = *batt highest cell* (V)
16. BLC = *batt lowest cell* (V)
17. BAC = *batt average cell* (V)
18. BDV = *batt delta voltage* (V)

Tabel 4.2 Sebagian Data Hasil Monitoring 1 Pada Hari Pertama

Jam	Data Hasil Monitoring								
	S	TP	SP	PV	P	PC	CC L	DC L	SO C
08:49:16	7	38,06 2	172,8 0	11 0		3,8 4	64	240	94
08:49:28	7	0,441	2,00	11 1		0,1 4	46	240	94
08:49:35	0	0,000	0,00	11 0		7,0 4	89	240	94
08:50:02	1	47,35 7	215,0 0	11 0		2,5 4	60	240	94
08:50:03	2	47,66 5	216,4 0	11 0		2,6 4	62	240	94
08:50:05	3	51,01 3	231,6 0	11 0		3,4 4	65	240	94
08:50:06	4	52,37 9	237,8 0	11 0		4,0 4	70	240	94
08:50:07	4	45,59 5	207,0 0	11 0		2,3 4	62	240	94

08:50:0 8	5	38,32 6	174,0 0	11 0		1,9 4	59	240	94
08:50:0 9	5	31,05 7	141,0 0	11 0		1,6 4	56	240	94
08:50:1 0	5	27,51 1	124,9 0	11 0		2,0 4	57	240	94

Tabel 4.3 Sebagian Data Hasil Monitoring 2 Pada Hari Pertama

Jam	Data Hasil Monitoring								
	AH	12 v	BH T	BL T	BA T	BH C	BL C	BA C	BD V
08:49: 16	113 ,9	11, 6	35	32	33	4,1 6	4,0 3	4,1 0	0,1 23
08:49: 28	113 ,9	11, 6	35	32	33	4,1 7	4,1 0	4,1 2	0,0 67
08:49: 35	113 ,9	11, 5	35	32	33	4,1 4	3,9 6	4,0 8	0,1 79
08:50: 02	113 ,9	11, 6	35	32	33	4,1 6	4,0 5	4,1 0	0,1 05
08:50: 03	113 ,8	11, 6	35	32	33	4,1 6	4,0 6	4,1 0	0,1 04
08:50: 05	113 ,9	11, 6	35	32	33	4,1 6	4,0 3	4,1 0	0,1 2
08:50: 06	113 ,9	11, 6	35	32	33	4,1 5	4,0 2	4,1 0	0,1 29
08:50: 07	113 ,9	11, 6	35	32	33	4,1 6	4,0 6	4,1 0	0,0 99
08:50: 08	113 ,9	11, 6	35	32	33	4,1 6	4,0 7	4,1 1	0,0 9
08:50: 09	113 ,9	11, 6	35	32	33	4,1 6	4,0 8	4,1 1	0,0 83
08:50: 10	113 ,9	11, 6	35	32	33	4,1 6	4,0 8	4,1 1	0,0 83

Hasil data monitoring diatas ditujukan untuk melihat konsumsi energi yang terpakai, energi yang tersedia, dan juga melihat energi yang masuk dalam baterai. Data tersebut merupakan faktor-faktor untuk menaikkan atau menurunkan kecepatan mobil Widya Wahana V dan akan dilihat melalui laptop pada mobil *escort* yang berada tepat dibelakang mobil Widya Wahana V. Standart yang dipakai untuk memberi instruksi kepada *driver* yaitu tim yang berada pada mobil *escort* akan menentukan apakah mobil Widya Wahana V akan bertambah kecepatanya atau berkurang kecepatannya dari faktor-faktor yang telah ditentukan. Informasi penambahan atau pengurangan kecepatan disampaikan kepada *driver* mobil Widya Wahana V dari tim dimobil *escort*.

4.5 Analisa Sistem Monitoring

Sistem monitoring pada mobil Widya Wahana V terdapat kekurangan atau kelemahan dalam keseluruhan sistem yang terjadi. Sebelum memasuki analisa pada sistem monitoring, analisa pertama yang dilakukan yaitu perbandingan antara mobil Widya Wahana tanpa sistem monitoring dengan mobil Widya Wahana menggunakan sistem monitoring. Pada mobil tanpa sistem monitoring akan susah untuk memanajemen energi dikarenakan tidak mengetahui konsumsi energi ataupun energi yang masuk pada saat mobil tersebut berjalan sehingga pada perlombaan 2013 jarak yang ditempuh 784 km, sedangkan pada perlombaan 2015 didapat 1647 km.

Tabel 4.4 Analisa Sistem Monitoring

Sistem Widya Wahana V	Analisa	Solusi
Pengambilan data kontroller	Sistem sudah bagus	-
Pengambilan data BMS	Terdapat gangguan <i>address</i>	Dibuatkan <i>board</i> pengambil data BMS
Pengambilan data	Arus tiap MPPT	Penambahan sensor

MPPT	belum terbaca	tiap MPPT dan dibuatkan <i>board</i> pengambil data dari sensor
<i>Board</i> joulemeter	Sistem sudah bagus	-
<i>Discovery board</i> STM32F3	Dimensi masih terlalu besar (6,6 cm x 9,1 cm)	Dibuatkan <i>board</i> dengan dimensi 6 cm x 5 cm
Sistem penampilan data	Sistem sudah bagus	-
Sistem penyimpanan data	Terdapat data yang tidak tersimpan	Penambahan komponen Micro SD pada <i>board</i> Xbee

Analisa berikutnya merupakan analisa pada sistem monitoring. Analisa pertama yaitu dimensi *discovery board* STM32F3 yang terlalu besar. Dengan adanya *discovery board* STM32F3 ini program yang digunakan dapat bermacam-macam dan dapat diisi dengan banyak program, namun dalam sistem monitoring ini tidaklah begitu banyak macam programnya, maka alangkah baiknya *discovery board* STM32F3 ini diganti dengan membuat *board* sendiri yang lebih kecil dan sederhana dengan memasang mikro STM32F3 sesuai dengan kebutuhan dalam sistem agar dapat mengefisiensikan fungsi (program) dan dimensi *board* mikrokontroler.



Gambar 4.24 STM32F3 Discovery Board

Analisa kedua yaitu sistem pembacaan arus tiap sel MPPT. Sistem monitoring pada mobil Widya Wahana V hanya menampilkan rata-rata data arus MPPT saja, sedangkan yang dibutuhkan yaitu data arus tiap MPPTnya. Agar dapat mengetahui kinerja atau arus yang keluar dari tiap MPPT, kabel *output* tiap MPPT diberi 1 sensor arus (LEM DHAB) atau dapat dikatakan 1 MPPT perlu 1 sensor arus dan juga perlu dibuatkan *board* tersendiri untuk mengambil dan mengolah data arus dengan begitu arus yang keluar tiap MPPT dapat terbaca atau tertampil.

Analisa ketiga yaitu sistem penyimpanan data. Penyimpanan data saat perlombaan *World Solar Car 2015* pada tim ITS ini melalui pengiriman data dari mobil Widya Wahana V dan data diterima oleh mobil *escort* kemudian data tersebut disimpan pada laptop. Kekhawatiran penyimpanan data seperti saat perlombaan *World Solar Car 2015* yaitu saat pengiriman data terputus secara tiba-tiba yang mengakibatkan penyimpanan data ikut terputus dan data yang tersimpan tidak keseluruhan. Agar data yang didapat atau disimpan secara sempurna, maka *board* XBEE pada mobil Widya Wahana V diberi slot untuk *micro SD* atau *SD card* dan pada *board* XBEE diberi tambahan program untuk penyimpanan data pada *micro SD*.



Gambar 4.25 Micro SD

Analisa empat yaitu kinerja baterai tiap selnya. Mobil Widya Wahana V terdapat sistem baterai yang terpecah dalam 2 box (bagian kiri dan kanan mobil) dan tiap box terdapat 14 sel. Dalam sistem monitoring ini belum dapat melihat kinerja baterai tiap sel pada saat mobil Widya Wahana V berjalan, dikarenakan terdapat gangguan pada *address* CAN BMS sehingga data yang didapat hanya sebagian saja. Untuk mengatasi gangguan tersebut, alangkah baiknya pengambilan data pada BMS ini dibuatkan *board* sendiri yang tidak tergabung dengan pengambilan data yang lain dan pada *board* tersebut diberi mikro STM32F3 pula.

LAMPIRAN

LAMPIRAN 1 Program Baca Data BMS

```
/* Define to prevent recursive inclusion -----
-----*/
#ifndef __MAIN_H
#define __MAIN_H
/* Includes -----
-----*/
#include "ch.h"
#include "hal.h"
#include <string.h>
#include "stm32f30x.h"
#include "usbcfg.h"
#include "chprintf.h"
/* Exported types -----
-----*/
/* Exported constants -----
-----*/
#define CANx                CAN1
#define CAN_CLK              RCC_APB1Periph_CAN1
#define CAN_RX_PIN           GPIO_Pin_0
#define CAN_TX_PIN           GPIO_Pin_1
#define CAN_GPIO_PORT        GPIO_D
#define CAN_GPIO_CLK         RCC_AHBPeriph_GPIOD
#define CAN_AF_PORT           GPIO_AF_7
#define CAN_RX_SOURCE         GPIO_PinSource0
#define CAN_TX_SOURCE         GPIO_PinSource1
//struktur komunikasi serial melalui USB
SerialUSBDriver SDU1;
BaseSequentialStream* chp =(BaseSequentialStream*)
&SDU1;
```

```

#define usb_lld_connect_bus(usbp)
#define usb_lld_disconnect_bus(usbp)
//definisi CAN Transmit
CanTxMsg TxMessage      = {0};
//definisi CAN Receive
CanRxMsg RxMessage      = {0};
CanRxMsg RxMessage44    = {0};
CanRxMsg RxMessage22    = {0};
CanRxMsg RxMessage1B    = {0};
CanRxMsg RxMessage3B    = {0};
CanRxMsg RxMessage6B2   = {0};
CanRxMsg RxMessage3CB   = {0};
CanRxMsg RawMessage     = {0};
int lowvolt1            = 0;
int lowvolt2            = 0;
int highvolt1           = 0;
int highvolt2           = 0;
int packvolt1           = 0;
int packvolt2           = 0;
int packcurrent1        = 0;
int packcurrent2        = 0;
int amphours1           = 0;
int amphours2           = 0;
int packDODraw          = 0;
int avgcellvolt1        = 0;
int avgcellvolt2        = 0;
int avgcellvoltraw      = 0;
int supply12vraw        = 0;
int amphoursraw         = 0;
int packcurrentraw      = 0;
int packsocraw          = 0;
int packvoltraw         = 0;

```

```

int lowvolt          = 0;
int highvolt         = 0;
float avgcellvolt     = 0.0000;
float deltacellvolt   = 0.0;
float packDOD         = 0.0;
float supply12v       = 0.0;
float amphours        = 0.0;
float packcurrent     = 0.0;
float packvolt        = 0.0;
float lowvoltcell     = 0.0000;
float highvoltcell    = 0.0000;
float packsoc         = 0.0;
int avgtemp           = 0;
int hightemp          = 0;
int lowtemp           = 0;
int packDCL           = 0;
int packCCL           = 0;
/* Exported macro -----
-----*/
/* Exported functions -----
----- */
static void CAN_Config(void);
#endif /* __MAIN_H */

```

LAMPIRAN 2 Program Pengolahan Data MPPT

```

/* Define to prevent recursive inclusion -----
-----*/
#ifndef __MAIN_H
#define __MAIN_H
/* Includes -----
-----*/
#include "ch.h"
#include "hal.h"
#include <string.h>
#include "stm32f30x.h"
#include "usbcfg.h"
#include "chprintf.h"
#include "adc_it.h"
#include "my_process.h"
#include "my_pin.h"
/* Exported types -----
-----*/
/* Exported constants -----
-----*/
#define CANx                CAN1
#define CAN_CLK              RCC_APB1Periph_CAN1
#define CAN_RX_PIN           GPIO_Pin_0
#define CAN_TX_PIN           GPIO_Pin_1
#define CAN_GPIO_PORT        GPIO_D
#define CAN_GPIO_CLK         RCC_AHBPeriph_GPIOD
#define CAN_AF_PORT           GPIO_AF_7
#define CAN_RX_SOURCE         GPIO_PinSource0
#define CAN_TX_SOURCE         GPIO_PinSource1
//struktur komunikasi serial melalui USB
SerialUSBDriver SDU1;

```

```

BaseSequentialStream* chp =(BaseSequentialStream*)
&SDU1;
#define usb_llid_connect_bus(usbp)
#define usb_llid_disconnect_bus(usbp)
//definisi CAN Transmit
CanTxMsg TxMessage      = {0};
CanTxMsg TxMessage2     = {0};
CanTxMsg TxMessage3     = {0};
CanTxMsg TxMessage4     = {0};
//definisi CAN Receive
CanRxMsg RxMessage      = {0};
CanRxMsg RxMessage44    = {0};
CanRxMsg RxMessage22    = {0};
CanRxMsg RxMessage1B    = {0};
CanRxMsg RxMessage3B    = {0};
CanRxMsg RxMessage6B2   = {0};
CanRxMsg RxMessage3CB   = {0};
CanRxMsg RawMessage     = {0};
int lowvolt1            = 0;
int lowvolt2            = 0;
int highvolt1           = 0;
int highvolt2           = 0;
int packvolt1           = 0;
int packvolt2           = 0;
int packcurrent1        = 0;
int packcurrent2        = 0;
int amphours1           = 0;
int amphours2           = 0;
int packDODraw          = 0;
int avgcellvolt1        = 0;
int avgcellvolt2        = 0;
int avgcellvoltraw      = 0;

```



```

int supply12vraw      = 0;
int amphoursraw      = 0;
int packcurrentraw    = 0;
int packsocraw       = 0;
int packvoltraw      = 0;
int lowvolt          = 0;
int highvolt         = 0;
float avgcellvolt     = 0.0000;
float deltacellvolt   = 0.0;
float packDOD         = 0.0;
float supply12v       = 0.0;
float amphours        = 0.0;
float packcurrent     = 0.0;
float packvolt        = 0.0;
float lowvoltcell     = 0.0000;
float highvoltcell    = 0.0000;
float packsoc         = 0.0;
int avgtemp           = 0;
int hightemp          = 0;
int lowtemp           = 0;
int packDCL           = 0;
int packCCL           = 0;
/* Exported macro -----
-----*/
/* Exported functions -----
----- */
static void CAN_Config(void);
extern adcsample_t adc_val;
extern adcsample_t adc_val2;
extern adcsample_t adc_val3;
extern adcsample_t adc_val4;
extern adcsample_t adc_val5;

```

```
extern adcsample_t adc_val6;
extern adcsample_t adc_val7;
extern adcsample_t adc_val8;
extern adcsample_t adc_val9;
extern adcsample_t adc_val10;
extern adcsample_t adc_val11;
extern adcsample_t adc_val12;
extern adcsample_t adc_val13;
extern adcsample_t adc_val14;
extern adcsample_t adc_val15;
extern adcsample_t adc_val16;
extern int mppt1_i_adc_rata;
extern int mppt2_i_adc_rata;
extern int mppt3_i_adc_rata;
extern int mppt4_i_adc_rata;
extern int mppt5_i_adc_rata;
extern int mppt6_i_adc_rata;
extern int mppt7_i_adc_rata;
extern int mppt8_i_adc_rata;
extern int mppt9_i_adc_rata;
extern int mppt10_i_adc_rata;
extern int mppt11_i_adc_rata;
extern int mppt12_i_adc_rata;
extern int mppt13_i_adc_rata;
extern int mppt14_i_adc_rata;
extern int mppt15_i_adc_rata;
extern float mppt1_i;
extern float mppt2_i;
extern float mppt3_i;
extern float mppt4_i;
extern float mppt5_i;
extern float mppt6_i;
```

```
extern float mppt7_i;  
extern float mppt8_i;  
extern float mppt9_i;  
extern float mppt10_i;  
extern float mppt11_i;  
extern float mppt12_i;  
extern float mppt13_i;  
extern float mppt14_i;  
extern float mppt15_i;  
#endif /* __MAIN_H */
```

LAMPIRAN 3 Program Joulemeter

```

/* Define to prevent recursive inclusion -----
-----*/
#ifndef __MAIN_H
#define __MAIN_H
/* Includes -----
-----*/
#include "ch.h"
#include "hal.h"
#include <string.h>
#include "stm32f30x.h"
#include "usbcfg.h"
#include "chprintf.h"
#include "adc_it.h"
#include "my_kirim.h"
#include "my_process.h"
#include "my_pin.h"
/* Exported types -----
-----*/
/* Exported constants -----
-----*/
#define CANx                CAN1
#define CAN_CLK              RCC_APB1Periph_CAN1
#define CAN_RX_PIN           GPIO_Pin_0
#define CAN_TX_PIN           GPIO_Pin_1
#define CAN_GPIO_PORT        GPIO_D
#define CAN_GPIO_CLK         RCC_AHBPeriph_GPIOD
#define CAN_AF_PORT           GPIO_AF_7
#define CAN_RX_SOURCE         GPIO_PinSource0
#define CAN_TX_SOURCE         GPIO_PinSource1
//struktur komunikasi serial melalui USB

```

```

SerialUSBDriver SDU1;
BaseSequentialStream* chp =(BaseSequentialStream*)
&SDU1;
#define usb_lld_connect_bus(usbp)
#define usb_lld_disconnect_bus(usbp)
//definisi CAN Transmit
CanTxMsg TxMessage      = {0};
CanTxMsg TxMessage2     = {0};
CanTxMsg TxMessage3     = {0};
CanTxMsg TxMessage4     = {0};
CanRxMsg RawMessage     = {0};
CanRxMsg RxMessage      = {0};
//definisi CAN Receive
//ID Tegangan
CanRxMsg RxMessage33    = {0};
CanRxMsg RxMessage44    = {0};
CanRxMsg RxMessage55    = {0};
CanRxMsg RxMessage66    = {0};
//ID Arus
CanRxMsg RxMessage77    = {0};
CanRxMsg RxMessage88    = {0};
CanRxMsg RxMessage99    = {0};
CanRxMsg RxMessage11    = {0};
//ID BMS
CanRxMsg RxMessage1B    = {0};
CanRxMsg RxMessage3B    = {0};
CanRxMsg RxMessage6B2   = {0};
CanRxMsg RxMessage3CB   = {0};
int lowvolt1            = 0;
int lowvolt2            = 0;
int highvolt1           = 0;
int highvolt2           = 0;

```

```

int packvolt1      = 0;
int packvolt2      = 0;
int packcurrent1   = 0;
int packcurrent2   = 0;
int amphours1      = 0;
int amphours2      = 0;
int packDODDraw    = 0;
int avgcellvolt1   = 0;
int avgcellvolt2   = 0;
int avgcellvoltraw = 0;
int supply12vraw   = 0;
int amphoursraw    = 0;
int packcurrentraw = 0;
int packsocraw     = 0;
int packvoltraw    = 0;
int lowvolt        = 0;
int highvolt       = 0;
float avgcellvolt   = 0.0000;
float deltacellvolt = 0.0;
float packDOD       = 0.0;
float supply12v     = 0.0;
float amphours      = 0.0;
float packcurrent    = 0.0;
float packvolt      = 0.0;
float lowvoltcell   = 0.0000;
float highvoltcell  = 0.0000;
float packsoc       = 0.0;
int avgtemp         = 0;
int hightemp        = 0;
int lowtemp         = 0;
int packDCL         = 0;
int packCCL         = 0;

```

```
//definisi data tegangan
```

```
int voltMPPT1_1    = 0;
int voltMPPT1_2    = 0;
int voltMPPT2_1    = 0;
int voltMPPT2_2    = 0;
int voltMPPT3_1    = 0;
int voltMPPT3_2    = 0;
int voltMPPT4_1    = 0;
int voltMPPT4_2    = 0;
int voltMPPT5_1    = 0;
int voltMPPT5_2    = 0;
int voltMPPT6_1    = 0;
int voltMPPT6_2    = 0;
int voltMPPT7_1    = 0;
int voltMPPT7_2    = 0;
int voltMPPT8_1    = 0;
int voltMPPT8_2    = 0;
int voltMPPT9_1    = 0;
int voltMPPT9_2    = 0;
int voltMPPT10_1   = 0;
int voltMPPT10_2   = 0;
int voltMPPT11_1   = 0;
int voltMPPT11_2   = 0;
int voltMPPT12_1   = 0;
int voltMPPT12_2   = 0;
int voltMPPT13_1   = 0;
int voltMPPT13_2   = 0;
int voltMPPT15_1   = 0;
int voltMPPT15_2   = 0;
int voltMPPT16_1   = 0;
int voltMPPT16_2   = 0;
int voltMPPT1      = 0;
```

```

int voltMPPT2      = 0;
int voltMPPT3      = 0;
int voltMPPT4      = 0;
int voltMPPT5      = 0;
int voltMPPT6      = 0;
int voltMPPT7      = 0;
int voltMPPT8      = 0;
int voltMPPT9      = 0;
int voltMPPT10     = 0;
int voltMPPT11     = 0;
int voltMPPT12     = 0;
int voltMPPT13     = 0;
int voltMPPT15     = 0;
int voltMPPT16     = 0;
    float mppt1_v ;
    float mppt2_v ;
    float mppt3_v ;
    float mppt4_v ;
    float mppt5_v ;
    float mppt6_v ;
    float mppt7_v ;
    float mppt8_v ;
    float mppt9_v ;
    float mppt10_v ;
    float mppt11_v ;
    float mppt12_v ;
    float mppt13_v ;
    float mppt15_v ;
    float mppt16_v ;

//definisi data arus
int currentMPPT1_1 = 0;

```



```

int currentMPPT1_2      = 0;
int currentMPPT2_1      = 0;
int currentMPPT2_2      = 0;
int currentMPPT3_1      = 0;
int currentMPPT3_2      = 0;
int currentMPPT4_1      = 0;
int currentMPPT4_2      = 0;
int currentMPPT5_1      = 0;
int currentMPPT5_2      = 0;
int currentMPPT6_1      = 0;
int currentMPPT6_2      = 0;
int currentMPPT7_1      = 0;
int currentMPPT7_2      = 0;
int currentMPPT8_1      = 0;
int currentMPPT8_2      = 0;
int currentMPPT9_1      = 0;
int currentMPPT9_2      = 0;
int currentMPPT10_1     = 0;
int currentMPPT10_2     = 0;
int currentMPPT11_1     = 0;
int currentMPPT11_2     = 0;
int currentMPPT12_1     = 0;
int currentMPPT12_2     = 0;
int currentMPPT13_1     = 0;
int currentMPPT13_2     = 0;
int currentMPPT14_1     = 0;
int currentMPPT14_2     = 0;
int currentMPPT15_1     = 0;
int currentMPPT15_2     = 0;
int currentMPPT1        = 0;
int currentMPPT2        = 0;
int currentMPPT3        = 0;

```

```

int currentMPPT4           = 0;
int currentMPPT5           = 0;
int currentMPPT6           = 0;
int currentMPPT7           = 0;
int currentMPPT8           = 0;
int currentMPPT9           = 0;
int currentMPPT10          = 0;
int currentMPPT11          = 0;
int currentMPPT12          = 0;
int currentMPPT13          = 0;
int currentMPPT14          = 0;
int currentMPPT15          = 0;
    float mppt1_i ;
    float mppt2_i ;
    float mppt3_i ;
    float mppt4_i ;
    float mppt5_i ;
    float mppt6_i ;
    float mppt7_i ;
    float mppt8_i ;
    float mppt9_i ;
    float mppt10_i ;
    float mppt11_i ;
    float mppt12_i ;
    float mppt13_i ;
    float mppt14_i ;
    float mppt15_i ;
extern float current1;
extern float current2;
extern float current3;
extern float voltage1;
extern float voltage2;

```

```

extern float voltage3;
extern float daya1;
extern float daya2;
extern float daya3;
extern float energy1_wh;
extern float energy2_wh;
extern float energy3_wh;
int arus1;
int arus2;
int arus3;
int volt1;
int volt2;
int volt3;
int watt1;
int watt2;
int watt3;
int joule1;
int joule2;
int joule3;
extern int milidetik,detik,menit,jam;
/* Exported macro -----
-----*/
/* Exported functions -----
----- */
static void CAN_Config(void);
#endif /* __MAIN_H */

```

LAMPIRAN 4 Program Display LCD

```

/* Define to prevent recursive inclusion -----
-----*/
#ifndef __MAIN_H
#define __MAIN_H
/* Includes -----
-----*/
#include "ch.h"
#include "hal.h"
#include <string.h>
#include "stm32f30x.h"
#include "usbcfg.h"
#include "chprintf.h"
/* Exported types -----
-----*/
/* Exported constants -----
-----*/
#define CANx                CAN1
#define CAN_CLK              RCC_APB1Periph_CAN1
#define CAN_RX_PIN           GPIO_Pin_0
#define CAN_TX_PIN           GPIO_Pin_1
#define CAN_GPIO_PORT        GPIO_D
#define CAN_GPIO_CLK         RCC_AHBPeriph_GPIOD
#define CAN_AF_PORT          GPIO_AF_7
#define CAN_RX_SOURCE         GPIO_PinSource0
#define CAN_TX_SOURCE         GPIO_PinSource1
//struktur komunikasi serial melalui USB
SerialUSBDriver SDU1;
BaseSequentialStream* chp =(BaseSequentialStream*)
&SDU1;
#define usb_llc_connect_bus(usbp)

```

```

#define usb_lld_disconnect_bus(usbp)
//definisi CAN Transmit
CanTxMsg TxMessage      = {0};
CanRxMsg RxMessage      = {0};
CanRxMsg RawMessage     = {0};
//definisi CAN Receive
//ID Sevcon
CanRxMsg RxMessage438   = {0};
//ID BMS
CanRxMsg RxMessage1B    = {0};
CanRxMsg RxMessage3B    = {0};
CanRxMsg RxMessage6B2   = {0};
CanRxMsg RxMessage3CB   = {0};
//definisi rpm Sevcon
int rpm1                 = 0;
int rpm2                 = 0;
int rpm                  = 0;
float speed               = 0.00;
//definisi data BMS
int lowvolt1             = 0;
int lowvolt2             = 0;
int highvolt1            = 0;
int highvolt2            = 0;
int packvolt1            = 0;
int packvolt2            = 0;
int packcurrent1         = 0;
int packcurrent2         = 0;
int amphours1            = 0;
int amphours2            = 0;
int packDODDraw          = 0;
int avgcellvolt1         = 0;
int avgcellvolt2         = 0;

```

```

int avgcellvoltraw    = 0;
int supply12vraw      = 0;
int amphoursraw       = 0;
int packcurrentraw    = 0;
int packsocraw        = 0;
int packvoltraw       = 0;
int lowvolt           = 0;
int highvolt          = 0;
float avgcellvolt      = 0.0000;
float deltacellvolt    = 0.0;
float packDOD          = 0.0;
float supply12v        = 0.0;
float amphours         = 0.0;
float packcurrent      = 0.0;
float packvolt         = 0.0;
float lowvoltcell      = 0.0000;
float highvoltcell     = 0.0000;
float packsoc          = 0.0;
int avgtemp            = 0;
int hightemp           = 0;
int lowtemp            = 0;
int packDCL            = 0;
int packCCL            = 0;
/* Exported macro -----
-----*/
/* Exported functions -----
----- */
static void CAN_Config(void);
#endif /* __MAIN_H */

```

LAMPIRAN 5 Program Untuk Display Monitoring Widya Wahana V

```
#include "interfacing.h"
#include "ui_interfacing.h"
#include "dial_display.h"
#include <iostream>
#include <fstream>
#include <unistd.h>
#include "ctime"
#include <QtGui>
#include <qwt_round_scale_draw.h>
```

```
using namespace std;
char date1;
float speed_sp = 0;
float speed_sp_persen = 0;
int baterai_persen = 0;
float mppt1_v=0;
float mppt2_v=0;
float mppt3_v=0;
float mppt4_v=0;
float mppt5_v=0;
float mppt6_v=0;
float mppt7_v=0;
float mppt8_v=0;
float mppt9_v=0;
float mppt10_v=0;
float mppt11_v=0;
float mppt12_v=0;
float mppt13_v=0;
float mppt14_v=0;
float mppt15_v=0;
float mppt1_i=0;
```

```
float mppt2_i=0;
float mppt3_i=0;
float mppt4_i=0;
float mppt5_i=0;
float mppt6_i=0;
float mppt7_i=0;
float mppt8_i=0;
float mppt9_i=0;
float mppt10_i=0;
float mppt11_i=0;
float mppt12_i=0;
float mppt13_i=0;
float mppt14_i=0;
float mppt15_i=0;
float bms_pack_soc=0;
float bms_pack_dcl=0;
float bms_pack_ccl=0;
float bms_avg_cellvoltage=0;
float bms_lowest_cellvoltage=0;
float bms_highest_cellvoltage=0;
float bms_delta_cellvoltage=0;
float bms_avg_battery_temperature=0;
float bms_lowest_battery_temperature=0;
float bms_highest_battery_temperature=0;
float bms_pack_ampereout=0;
float bms_pack_voltageout=0;
float bms_pack_amperehour=0;
float bms_12vsupply=0;
float bms_runtime=0;
float bms_v_cell1=0;
float bms_v_cell2=0;
float bms_v_cell3=0;
```



```
float bms_v_cell4=0;
float bms_v_cell5=0;
float bms_v_cell6=0;
float bms_v_cell7=0;
float bms_v_cell8=0;
float bms_v_cell9=0;
float bms_v_cell10=0;
float bms_v_cell11=0;
float bms_v_cell12=0;
float bms_v_cell13=0;
float bms_v_cell14=0;
float bms_v_cell15=0;
float bms_v_cell16=0;
float bms_v_cell17=0;
float bms_v_cell18=0;
float bms_v_cell19=0;
float bms_v_cell20=0;
float bms_v_cell21=0;
float bms_v_cell22=0;
float bms_v_cell23=0;
float bms_v_cell24=0;
float bms_v_cell25=0;
float bms_v_cell26=0;
float bms_v_cell27=0;
float bms_v_cell28=0;
float bms_v_cell29=0;
float bms_v_cell30=0;
float bms_v_cell31=0;
float bms_v_cell32=0;
float bms_v_cell33=0;
float panel1_current=0;
float panel1_voltage=0;
```

```
float panel1_power=0;
float panel1_energy=0;
float panel2_current=0;
float panel2_current1=0;
float panel2_voltage=0;
float panel2_power=0;
float panel2_energy=0;
float panel3_current=0;
float panel3_voltage=0;
float panel3_power=0;
float panel3_energy=0;
int milidetik = 0,detik=0,menit = 0,jam = 0;
float controller_current=0;
float controller_voltage=0;
float controller_power=0;
float controller_energy=0;
float energy_input=0;
float energy_output=0;
float energy_terpakai=0;
float energy_terserap=0;
float current_input=0;
float current_output=0;
float total_power_in=0;
float total_power_out=0;
float ampereout = 0;
float ampereout_avg=0;
float voltage_avg =0;
float speed_avg =0;
float a = 6553.5;
float speed =0;
int hitung = 0;
int i=0;
```

```

int vcount;
int vbase;
int data_index=0;

interfacing::interfacing(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::interfacing)
{
    ui->setupUi(this);

    ui->dateLabel->setText(QDate::currentDate().toString("dd.
MM.yyyy"));

    ui->timeLabel->setText(QTime::currentTime().toString());
    interfacing::setFixedSize(1100,650);
    ui->chk_data->setChecked(false);
    QPixmap pix2 ("/home/monitor/Documents/solar
car/program fix
interface/ITS_SOLARCAR_V1.57/panel.png");
    QPixmap newpix2 = pix2.scaled(QSize(1060,460));
    ui->lbl_overview->setPixmap(newpix2);
    QPixmap pix3 ("/home/monitor/Documents/solar
car/program fix
interface/ITS_SOLARCAR_V1.57/title.png");
    QPixmap newpix3 = pix3.scaled(QSize(1060,61));
    ui->lbl_title->setPixmap(newpix3);
    obj_timer = new QTimer(this);

    connect(obj_timer,SIGNAL(timeout()),this,SLOT(onTimeo
ut()));
    obj_timer->start(1);
    ui->tabWidget->addTab(new dial_display, "Display");

```

```

    ui->tabWidget->setAutoFillBackground(true);
    ui->txt_Terminal->installEventFilter(this);
    obj_port= new QSerialPort(this);

connect(obj_port,SIGNAL(readyRead()),SLOT(readData())
);

    vcount=0;
    ui->cmb_Baud->setCurrentIndex(1);
    //ui->cmb_Port->setCurrentText("pilih");

    ui->main_tbl->horizontalHeader()->setSectionResizeMode(
    QHeaderView::Stretch);

    ui->main_tbl->verticalHeader()->setSectionResizeMode(Q
    HeaderView::Stretch);
}
void interfacing::onTimeout()
{

    ui->timeLabel->setText(QTime::currentTime().toString("H:
    m:ss"));
}
interfacing::~~interfacing()
{
    obj_timer->stop();
    delete ui;
}
void interfacing::on_actionQuit_triggered()
{
    if(obj_port->isOpen()){obj_port->close();}
    QApplication::quit();
}

```

```

}
void interfacing::on_actionAboutQt_triggered()
{
    QApplication::aboutQt();
}
QVector<double>
datay1(1000),datay2(1000),datay3(1000);
void interfacing::readData(){
    QByteArray rawdata=obj_port->readAll();
    ui->txt_Terminal->insertPlainText(rawdata);
    ui->txt_Terminal_2->insertPlainText(rawdata);
    QString indata=ui->txt_Terminal->toPlainText();
    QStringList lsindata= indata.split("\n");
    uint cdata= lsindata.count();
    if(cdata<=2){return;}
    QStringList lsdata = lsindata[cdata-2].split(",");
    ui->txt_Terminal->clear();
    if(lsdata.size()<44 || lsdata.size()>44)
    {return;
        ui->chk_data->setChecked(false);}
    else{
        ui->chk_data->setChecked(true);
        speed = lsdata[0].toFloat();
        speed_sp = lsdata[1].toFloat();
        bms_pack_soc = lsdata[2].toFloat();
        bms_pack_dcl = lsdata[3].toFloat();
        bms_pack_ccl = lsdata[4].toFloat();
        bms_lowest_cellvoltage = lsdata[5].toFloat();
        bms_highest_cellvoltage = lsdata[6].toFloat();
        bms_highest_battery_temperature = lsdata[7].toFloat();
        bms_lowest_battery_temperature = lsdata[8].toFloat();
        bms_pack_ampereout = lsdata[9].toFloat()*0.1;
    }
}

```

```
bms_pack_voltageout = lsdata[10].toFloat();
bms_pack_amperehour = lsdata[11].toFloat();
bms_12vsupply = lsdata[12].toFloat();
bms_delta_cellvoltage = lsdata[13].toFloat();
bms_avg_cellvolttage = lsdata[14].toFloat();
panel1_current =lsdata[15].toFloat();
panel2_current =lsdata[16].toFloat();
panel3_current =lsdata[17].toFloat();
panel1_voltage =lsdata[18].toFloat();
panel2_voltage =lsdata[19].toFloat();
panel3_voltage =lsdata[20].toFloat();
milidetik = lsdata[21].toInt();
detik = lsdata[22].toInt();
menit = lsdata[23].toInt();
jam = lsdata[24].toInt();
bms_avg_battery_temperature = lsdata[25].toFloat();
mppt1_i = lsdata[26].toFloat();
mppt2_i = lsdata[27].toFloat();
mppt3_i = lsdata[28].toFloat();
mppt4_i = lsdata[29].toFloat();
mppt5_i = lsdata[30].toFloat();
mppt6_i = lsdata[31].toFloat();
mppt7_i = lsdata[32].toFloat();
mppt8_i = lsdata[33].toFloat();
mppt9_i = lsdata[34].toFloat();
mppt10_i = lsdata[35].toFloat();
mppt11_i = lsdata[36].toFloat();
mppt12_i = lsdata[37].toFloat();
mppt13_i = lsdata[38].toFloat();
mppt14_i = lsdata[39].toFloat();
mppt15_i = lsdata[40].toFloat();
ampereout_avg= lsdata[41].toFloat();
```

```
voltage_avg= lsdata[42].toFloat();
speed_avg= lsdata[43].toFloat();
//      bms_v_cell1 = lsdata[24].toFloat();
//      bms_v_cell2 = lsdata[25].toFloat();
//      bms_v_cell3 = lsdata[26].toFloat();
//      bms_v_cell4 = lsdata[27].toFloat();
//      bms_v_cell5 = lsdata[28].toFloat();
//      bms_v_cell6 = lsdata[29].toFloat();
//      bms_v_cell7 = lsdata[30].toFloat();
//      bms_v_cell8 = lsdata[31].toFloat();
//      bms_v_cell9 = lsdata[32].toFloat();
//      bms_v_cell10 = lsdata[33].toFloat();
//      bms_v_cell11 = lsdata[34].toFloat();
//      bms_v_cell12 = lsdata[35].toFloat();
//      bms_v_cell13 = lsdata[36].toFloat();
//      bms_v_cell14 = lsdata[37].toFloat();
//      bms_v_cell15 = lsdata[38].toFloat();
//      bms_v_cell16 = lsdata[39].toFloat();
//      bms_v_cell17 = lsdata[40].toFloat();
//      bms_v_cell18 = lsdata[41].toFloat();
//      bms_v_cell19 = lsdata[42].toFloat();
//      bms_v_cell20 = lsdata[44].toFloat();
//      bms_v_cell21 = lsdata[44].toFloat();
//      bms_v_cell22 = lsdata[45].toFloat();
//      bms_v_cell23 = lsdata[46].toFloat();
//      bms_v_cell24 = lsdata[47].toFloat();
//      bms_v_cell25 = lsdata[48].toFloat();
//      bms_v_cell26 = lsdata[49].toFloat();
//      bms_v_cell27 = lsdata[50].toFloat();
//      bms_v_cell28 = lsdata[51].toFloat();
//      bms_v_cell29 = lsdata[52].toFloat();
//      bms_v_cell30 = lsdata[53].toFloat();
```

```

//      bms_v_cell31 = lsdata[54].toFloat();
//      bms_v_cell32 = lsdata[55].toFloat();
//      bms_v_cell33 = lsdata[56].toFloat();
}
    datax[vcount]=vcount;
    datay1[vcount]=controller_current;
    datay2[vcount]=speed;
    datay3[vcount]=controller_voltage;
    vcount++;
    hitung++;
    vbase = vcount-100;
    ui->wgt_GraphPlot->addGraph();
    ui->wgt_GraphPlot->graph()->setData(datax,datay1);
    ui->wgt_GraphPlot->graph()->setPen(QPen(Qt::red));
    ui->wgt_GraphPlot->addGraph();
    ui->wgt_GraphPlot->graph()->setData(datax,datay2);
    ui->wgt_GraphPlot->graph()->setPen(QPen(Qt::yellow));
    ui->wgt_GraphPlot->addGraph();
    ui->wgt_GraphPlot->graph()->setData(datax,datay3);
    ui->wgt_GraphPlot->graph()->setPen(QPen(Qt::black));
    ui->wgt_GraphPlot->yAxis->setRange(-50,122);
    ui->wgt_GraphPlot->xAxis->setRange(vbase,vcount);
    ui->wgt_GraphPlot->setInteractions(QCP::iRangeDrag |
QCP::iRangeZoom);

ui->wgt_GraphPlot->axisRect()->setRangeDrag(Qt::Horizontal);
    ui->wgt_GraphPlot->replot();
    perhitungan();
    monitor_display();
    display_data();
    alur();

```



```

    }
void interfacing::on_btn_Open_clicked()
{
    QString dev_name="/dev/";
    dev_name += ui->cmb_Port->currentText();
    QSerialPort::BaudRate dev_baud;
    QSerialPort::DataBits dev_DataBits;
    QSerialPort::StopBits dev_StopBits;
    QSerialPort::FlowControl dev_FlowControl;
    QSerialPort::Parity dev_Parity;

    int baud_index=ui->cmb_Baud->currentIndex();
    int DataBits_index= ui->cmb_DataBits->currentIndex();
    int StopBits_index= ui->cmb_StopBits->currentIndex();
    int FlowControl_index=
ui->cmb_FlowControl->currentIndex();
    int Parity_index= ui->cmb_Parity->currentIndex();
    switch(baud_index){
    case 0: dev_baud=QSerialPort::Baud9600;break;
    case 1: dev_baud=QSerialPort::Baud38400;break;
    case 2: dev_baud=QSerialPort::Baud57600;break;
    case 3: dev_baud=QSerialPort::Baud115200;break;
    }
    switch(DataBits_index){
    case 0: dev_DataBits=QSerialPort::Data5;break;
    case 1: dev_DataBits=QSerialPort::Data6;break;
    case 2: dev_DataBits=QSerialPort::Data7;break;
    case 3: dev_DataBits=QSerialPort::Data8;break;
    }
    switch(StopBits_index){
    case 0: dev_StopBits=QSerialPort::OneStop;break;
    case 1:

```

```

dev_StopBits=QSerialPort::OneAndHalfStop;break;
    case 2: dev_StopBits=QSerialPort::TwoStop;break;
    }
    switch (FlowControl_index) {
    case 0:
dev_FlowControl=QSerialPort::NoFlowControl;break;
    case 1:
dev_FlowControl=QSerialPort::HardwareControl;break;
    case 2:
dev_FlowControl=QSerialPort::SoftwareControl;break;
    }
    switch(Parity_index){
    case 0: dev_Parity=QSerialPort::NoParity;break;
    case 1: dev_Parity=QSerialPort::EvenParity;break;
    case 2: dev_Parity=QSerialPort::OddParity;break;
    case 3: dev_Parity=QSerialPort::SpaceParity;break;
    case 4: dev_Parity=QSerialPort::MarkParity;break;
    }
    if(ui->btn_Open->text()=="Open"){
        obj_port->setPortName(dev_name);
        if(obj_port->open(QIODevice::ReadWrite)){
//
obj_port->setBaudRate(dev_baud,QSerialPort::AllDirection
s);
        obj_port->setBaudRate(dev_baud);
        obj_port->setDataBits(dev_DataBits);
        obj_port->setStopBits(dev_StopBits);
        obj_port->setFlowControl(dev_FlowControl);
        obj_port->setParity(dev_Parity);
        QMessageBox::information(this,"success","port
success on "+dev_name );
        ui->btn_Open->setText("Close");

```

```

        ui->txt_portname->setText(dev_name);
        ui->cmb_Baud->setEnabled(false);
        ui->cmb_DataBits->setEnabled(false);
        ui->cmb_FlowControl->setEnabled(false);
        ui->cmb_Parity->setEnabled(false);
        ui->cmb_Port->setEnabled(false);
        ui->cmb_StopBits->setEnabled(false);
    }
    else{
        QMessageBox::critical(this,"Failed","port failed on
"+dev_name);
        ui->btn_Open->setText("Open");
        ui->txt_portname->clear();
    }
    }else{if(obj_port->isOpen()){obj_port->close();
    }
    ui->btn_Open->setText("Open");
    ui->txt_portname->clear();
    ui->cmb_Baud->setEnabled(true);
    ui->cmb_DataBits->setEnabled(true);
    ui->cmb_FlowControl->setEnabled(true);
    ui->cmb_Parity->setEnabled(true);
    ui->cmb_Port->setEnabled(true);
    ui->cmb_StopBits->setEnabled(true);
    // ui->cmb_Baud->setCurrentIndex(4);
    // ui->cmb_Port->setCurrentText("pilih");
    }
}

bool interfacing::eventFilter(QObject *obj, QEvent *event){
    if(event->type()== QEvent::KeyPress){
        QKeyEvent* keyEvent =
        static_cast<QKeyEvent* >(event);

```

```

        if(obj_port->isOpen()){
            QString s = keyEvent->text();
            if(s.length()){
                char ch = s.at(0).toLatin1();
                QByteArray chb=
QByteArray::fromRawData(&ch,1);
                obj_port->write(chb);
                if(ui->chk_LocalEcho->isChecked()){
                    ui->txt_Terminal->insertPlainText(chb);
                }
            }
        }
        return true;
    }
    else{
        return QObject::eventFilter(obj, event);
    }
}

void interfacing::on_btn_Clear_clicked()
{
    ui->txt_Terminal->clear();
    ui->txt_Terminal_2->clear();
}

void interfacing::on_btn_ClearGraph_clicked()
{
    int i;
    for(i=0;i<1000;i++){
        datax[i]=0;
        datay1[i]=0;
        datay2[i]=0;
        datay3[i]=0;
    }
}

```

```

vcount=0;
if(obj_timer->isActive()){
    obj_timer->stop();
}
ui->wgt_GraphPlot->clearGraphs();
ui->wgt_GraphPlot->replot();
}
void interfacing::add_number_data(int vi, int vj, float vint,
QTableWidget *vtable){
    QTableWidgetItem *myitem = new QTableWidgetItem();
    myitem->setText(QString::number(vint));
    vtable->setItem(vi,vj,myitem);
}
void interfacing::display_data(){

add_number_data(data_index,0,total_power_in,ui->main_tb
l);

add_number_data(data_index,1,total_power_out,ui->main_t
bl);

add_number_data(data_index,2,energy_terserap,ui->main_t
bl);

add_number_data(data_index,3,energy_terpakai,ui->main_t
bl);
    add_number_data(data_index,4,speed,ui->main_tbl);

add_number_data(data_index,5,current_output,ui->main_tbl
);
    ui->main_tbl->setRowCount(data_index+2);
    data_index++;

```

```

}
void interfacing::alur()
{
    time_t rawtime;
    struct tm * timeinfo;
    char buffer [80];
    time (&rawtime);
    timeinfo = localtime (&rawtime);
    strftime (buffer,80,"%x,%T",timeinfo);
    puts(buffer);
    ofstream myfile;
    myfile.open("data.txt",ios::app);
    //  myfile
    <<buffer<<","<<speed<<","<<speed_sp<<","<<controller_
    voltage<<","<<controller_current<<","<<controller_power<
    <","<<controller_energy<<","
    //
    <<panel1_voltage<<","<<panel1_current<<","<<panel1_po
    wer<<","<<panel1_energy<<","
    //
    <<panel2_voltage<<","<<panel2_current<<","<<panel2_po
    wer<<","<<panel2_energy<<","
    //
    <<panel3_voltage<<","<<panel3_current<<","<<panel3_po
    wer<<","<<panel3_energy<<","
    //
    <<bms_pack_ccl<<","<<bms_pack_dcl<<","<<bms_pack_s
    oc<<","<<bms_pack_amperehour<<","
    //
    <<bms_12vsupply<<","<<bms_highest_battery_temperatur
    e<<","<<bms_lowest_battery_temperature<<","<<bms_avg
    _battery_temperature<<","

```

```

//
<<bms_highest_cellvoltage<<","<<bms_lowest_cellvoltage
<<","<<bms_avg_cellvoltage<<","<<bms_delta_cellvoltage<<","
//
<<mppt1_i<<","<<mppt2_i<<","<<mppt3_i<<","<<mppt4_i<<","<<mppt5_i<<","
//
<<mppt6_i<<","<<mppt7_i<<","<<mppt8_i<<","<<mppt9_i<<","<<mppt10_i<<","
//
<<mppt11_i<<","<<mppt12_i<<","<<mppt13_i<<","<<mppt14_i<<","<<mppt15_i;
    myfile
<<buffer<<","<<speed<<","<<speed_sp<<","<<bms_pack_voltageout<<","<<bms_pack_ampereout<<","<<ampereout<<","<<controller_current<<","
<<panel1_current<<","<<panel2_current1<<","<<panel3_current<<","<<ampereout_avg<<","<<speed_avg<<","<<voltage_avg<<","<<bms_pack_ccl<<","
<<bms_pack_dcl<<","<<bms_pack_soc<<","<<bms_pack_amperehour<<","
<<bms_12vsupply<<","<<bms_highest_battery_temperature<<","<<bms_lowest_battery_temperature<<","
<<bms_avg_battery_temperature<<","<<bms_highest_cellvoltage<<","<<bms_lowest_cellvoltage<<","
<<bms_avg_cellvoltage<<","<<bms_delta_cellvoltage<<","

```

"

```
<<mppt1_i<<","<<mppt2_i<<","<<mppt3_i<<","<<mppt4_
i<<","<<mppt5_i<<","
```

```
<<mppt6_i<<","<<mppt7_i<<","<<mppt8_i<<","<<mppt9_
i<<","<<mppt10_i<<","
```

```
<<mppt11_i<<","<<mppt12_i<<","<<mppt13_i<<","<<mp
pt14_i<<","<<mppt15_i;
```

```
myfile <<"\n";
```

```
myfile.flush();
```

```
myfile.close();
```

```
}
```

```
void interfacing::perhitungan(){
```

```
if(bms_pack_ampereout>6000){
```

```
(ampereout=bms_pack_ampereout-a);
```

```
controller_current=((ampereout-panel2_current1)-1);
```

```
}
```

```
else{
```

```
ampereout=bms_pack_ampereout;
```

```
controller_current=(ampereout-panel2_current1);
```

```
}
```

```
if(speed>6000){
```

```
(speed=0);
```

```
}
```

```
if(speed_avg>6000){
```

```
(speed_avg=0);
```

```
}
```

```
panel2_current1 = ((panel2_current-3143)/-50);
```

```
panel1_power = (panel1_current*bms_pack_voltageout);
```

```
panel2_power = (panel2_current*bms_pack_voltageout);
```



```

    panel3_power = (panel3_current*bms_pack_voltageout);
    total_power_in
=(panel1_power+panel2_power+panel3_power);
    total_power_out = (controller_power);
    controller_power =
(bms_pack_ampereout*bms_pack_voltageout);
    controller_voltage = bms_pack_voltageout;
    panel1_energy += (panel1_power/3600);
    panel2_energy += (panel2_power/3600);
    panel3_energy += (panel2_power/3600);
    controller_energy += (controller_power/3600);
    current_input =
(panel1_current+panel2_current+panel3_current);
    current_output = (controller_current);
    energy_input =
(panel1_energy+panel2_energy+panel3_energy);
    energy_output = (controller_energy);
    energy_terpakai += (controller_energy);
    energy_terserap += (energy_input);
    speed_sp_persen=((speed_sp/454)*100);
    baterai_persen = (((bms_pack_voltageout-
75.6)/37.8)*100);
}
void interfacing::monitor_display()
{

ui->txt_bms_lowest_battery_temperature->setText(QString:
:number(bms_lowest_battery_temperature));

ui->txt_bms_highest_battery_temperature->setText(QString
:number(bms_highest_battery_temperature));

```

```
ui->txt_bms_average_battery_temperature->setText(QString::number(bms_avg_battery_temperature));
```

```
ui->txt_bms_lowest_cellvoltage->setText(QString::number(bms_lowest_cellvoltage));
```

```
ui->txt_bms_highest_cellvoltage->setText(QString::number(bms_highest_cellvoltage));
```

```
ui->txt_bms_average_cellvoltage->setText(QString::number(bms_avg_cellvoltage));
```

```
ui->txt_bms_delta_cellvoltage->setText(QString::number(bms_delta_cellvoltage));
```

```
ui->txt_bms_pack_soc->setText(QString::number(bms_pack_soc));
```

```
ui->txt_bms_pack_dcl->setText(QString::number(bms_pack_dcl));
```

```
ui->txt_bms_pack_ccl->setText(QString::number(bms_pack_ccl));
```

```
ui->txt_bms_pack_ampereout->setText(QString::number(controller_current));
```

```
ui->txt_bms_pack_voltageout->setText(QString::number(bms_pack_voltageout));
```

```
ui->txt_bms_pack_amperehour->setText(QString::number(bms_pack_amperehour));
```

```
ui->txt_bms_12vsupply->setText(QString::number(bms_12
vsupply));
```

```
ui->txt_bms_runtime->setText(QString::number(bms_runti
me));
```

```
    ui->txt_mppt1_v->setText(QString::number(mppt1_v));
    ui->txt_mppt2_v->setText(QString::number(mppt2_v));
    ui->txt_mppt3_v->setText(QString::number(mppt3_v));
    ui->txt_mppt4_v->setText(QString::number(mppt4_v));
    ui->txt_mppt5_v->setText(QString::number(mppt5_v));
    ui->txt_mppt6_v->setText(QString::number(mppt6_v));
    ui->txt_mppt7_v->setText(QString::number(mppt7_v));
    ui->txt_mppt8_v->setText(QString::number(mppt8_v));
    ui->txt_mppt9_v->setText(QString::number(mppt9_v));
```

```
ui->txt_mppt10_v->setText(QString::number(mppt10_v));
```

```
ui->txt_mppt11_v->setText(QString::number(mppt11_v));
```

```
ui->txt_mppt12_v->setText(QString::number(mppt12_v));
```

```
ui->txt_mppt13_v->setText(QString::number(mppt13_v));
```

```
ui->txt_mppt14_v->setText(QString::number(mppt14_v));
```

```
ui->txt_mppt15_v->setText(QString::number(mppt15_v));
    ui->txt_mppt1_i->setText(QString::number(mppt1_i));
    ui->txt_mppt2_i->setText(QString::number(mppt2_i));
    ui->txt_mppt3_i->setText(QString::number(mppt3_i));
    ui->txt_mppt4_i->setText(QString::number(mppt4_i));
    ui->txt_mppt5_i->setText(QString::number(mppt5_i));
```

```
ui->txt_mppt6_i->setText(QString::number(mppt6_i));
ui->txt_mppt7_i->setText(QString::number(mppt7_i));
ui->txt_mppt8_i->setText(QString::number(mppt8_i));
ui->txt_mppt9_i->setText(QString::number(mppt9_i));
ui->txt_mppt10_i->setText(QString::number(mppt10_i));
ui->txt_mppt11_i->setText(QString::number(mppt11_i));
ui->txt_mppt12_i->setText(QString::number(mppt12_i));
ui->txt_mppt13_i->setText(QString::number(mppt13_i));
ui->txt_mppt14_i->setText(QString::number(mppt14_i));
ui->txt_mppt15_i->setText(QString::number(mppt15_i));

ui->txt_panel1_current->setText(QString::number(panel1_c
urrent));

ui->txt_panel1_voltage->setText(QString::number(bms_pac
k_voltageout));

ui->txt_panel1_power->setText(QString::number(panel1_po
wer));

ui->txt_panel1_energy->setText(QString::number(panel1_e
nergy));

ui->txt_panel2_current->setText(QString::number(panel2_c
urrent1));

ui->txt_panel2_voltage->setText(QString::number(bms_pac
k_voltageout));

ui->txt_panel2_power->setText(QString::number(panel2_po
wer));
```

```
ui->txt_panel2_energy->setText(QString::number(panel2_e
nergy));

ui->txt_panel3_current->setText(QString::number(panel3_c
urrent));

ui->txt_panel3_voltage->setText(QString::number(bms_pac
k_voltageout));

ui->txt_panel3_power->setText(QString::number(panel3_po
wer));

ui->txt_panel3_energy->setText(QString::number(panel3_e
nergy));

ui->txt_controller_current->setText(QString::number(amper
eout));

ui->txt_controller_voltage->setText(QString::number(bms_
pack_voltageout));

ui->txt_controller_power->setText(QString::number(control
ler_power));

ui->txt_controller_energy->setText(QString::number(contro
ller_energy));
    ui->txt_jam->setText(QString::number(jam));
    ui->txt_menit->setText(QString::number(menit));
    ui->txt_detik->setText(QString::number(detik));

ui->txt_energy_input->setText(QString::number(energy_inp
ut));
```

```
ui->txt_energy_output->setText(QString::number(energy_o  
utput));  
  
ui->txt_bms_cell1->setText(QString::number(bms_v_cell1))  
;  
  
ui->txt_bms_cell2->setText(QString::number(bms_v_cell2))  
;  
  
ui->txt_bms_cell3->setText(QString::number(bms_v_cell3))  
;  
  
ui->txt_bms_cell4->setText(QString::number(bms_v_cell4))  
;  
  
ui->txt_bms_cell5->setText(QString::number(bms_v_cell5))  
;  
  
ui->txt_bms_cell6->setText(QString::number(bms_v_cell6))  
;  
  
ui->txt_bms_cell7->setText(QString::number(bms_v_cell7))  
;  
  
ui->txt_bms_cell8->setText(QString::number(bms_v_cell8))  
;  
  
ui->txt_bms_cell9->setText(QString::number(bms_v_cell9))  
;  
  
ui->txt_bms_cell10->setText(QString::number(bms_v_cell1
```

```
0));
```

```
ui->txt_bms_cell11->setText(QString::number(bms_v_cell11));
```

```
ui->txt_bms_cell12->setText(QString::number(bms_v_cell12));
```

```
ui->txt_bms_cell13->setText(QString::number(bms_v_cell13));
```

```
ui->txt_bms_cell14->setText(QString::number(bms_v_cell14));
```

```
ui->txt_bms_cell15->setText(QString::number(bms_v_cell15));\
```

```
ui->txt_bms_cell16->setText(QString::number(bms_v_cell16));
```

```
ui->txt_bms_cell17->setText(QString::number(bms_v_cell17));
```

```
ui->txt_bms_cell18->setText(QString::number(bms_v_cell18));
```

```
ui->txt_bms_cell19->setText(QString::number(bms_v_cell19));
```

```
ui->txt_bms_cell20->setText(QString::number(bms_v_cell20));
```

```
ui->txt_bms_cell21->setText(QString::number(bms_v_cell2  
1));
```

```
ui->txt_bms_cell22->setText(QString::number(bms_v_cell2  
2));
```

```
ui->txt_bms_cell23->setText(QString::number(bms_v_cell2  
3));
```

```
ui->txt_bms_cell24->setText(QString::number(bms_v_cell2  
4));
```

```
ui->txt_bms_cell25->setText(QString::number(bms_v_cell2  
5));
```

```
ui->txt_bms_cell26->setText(QString::number(bms_v_cell2  
6));
```

```
ui->txt_bms_cell27->setText(QString::number(bms_v_cell2  
7));
```

```
ui->txt_bms_cell28->setText(QString::number(bms_v_cell2  
8));
```

```
ui->txt_bms_cell29->setText(QString::number(bms_v_cell2  
9));
```

```
ui->txt_bms_cell30->setText(QString::number(bms_v_cell3  
0));
```

```
ui->txt_bms_cell31->setText(QString::number(bms_v_cell3  
1));
```



```
ui->txt_bms_cell32->setText(QString::number(bms_v_cell32));

ui->txt_bms_cell33->setText(QString::number(bms_v_cell33));

ui->txt_panel1_current_disp->setText(QString::number(panel1_current));

ui->txt_panel2_current_disp->setText(QString::number(panel2_current));

ui->txt_panel3_current_disp->setText(QString::number(panel3_current));

ui->txt_paneltotal_current_disp->setText(QString::number(panel2_current1));

ui->txt_bms_peakvoltage_disp->setText(QString::number(controller_voltage));

ui->txt_bms_pack_ampereout_disp->setText(QString::number(ampereout));

ui->txt_controller_current_disp->setText(QString::number(controller_current));

ui->txt_speed_disp->setText(QString::number(speed));

ui->txt_speedsp_disp->setText(QString::number(speed_sp));
;
```

```
ui->txt_speedsp_disp_2->setText(QString::number(speed_s
p_persen));
```

```
ui->txt_bms_sisabaterai->setText(QString::number(baterai_
persen));
```

```
ui->txt_voltageoutavg->setText(QString::number(voltage_a
vg));
```

```
ui->txt_velocityavg->setText(QString::number(speed_avg));
```

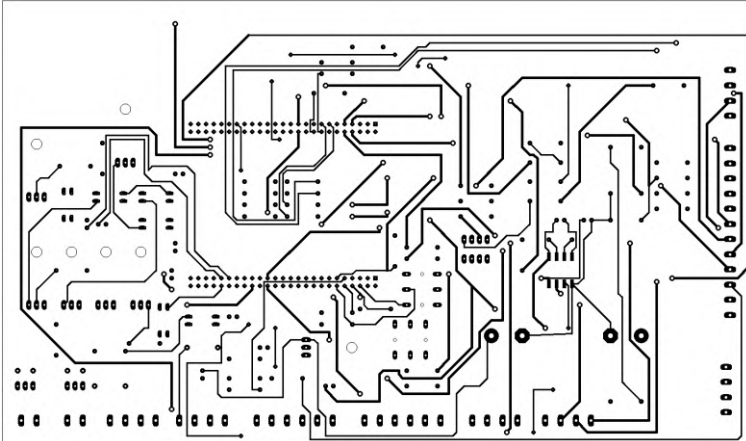
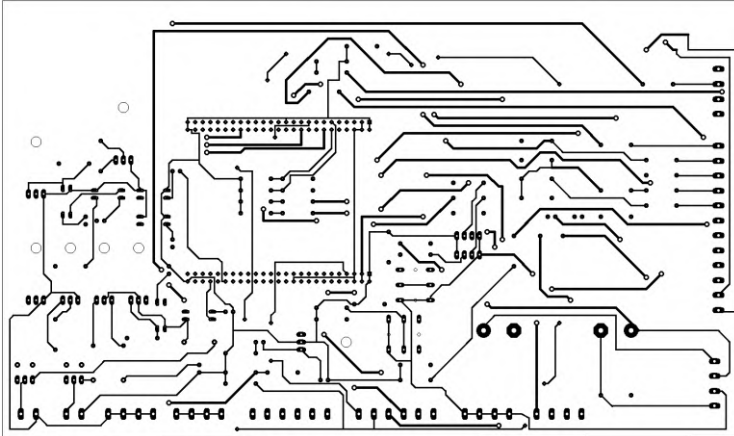
```
ui->txt_ampereoutavg->setText(QString::number(ampereout
_avg));
}
```

```
void interfacing::on_lbl_gambarits_linkActivated(const
QString &link)
```

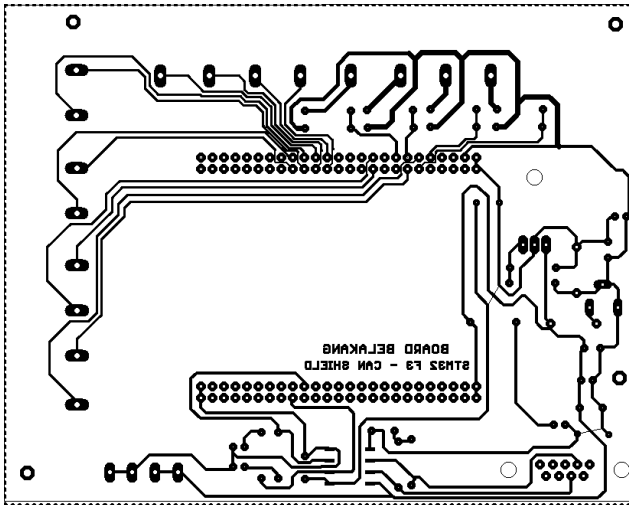
```
{
{return;}
}
```

```
void interfacing::on_lbl_overview_linkActivated(const
QString &link)
```

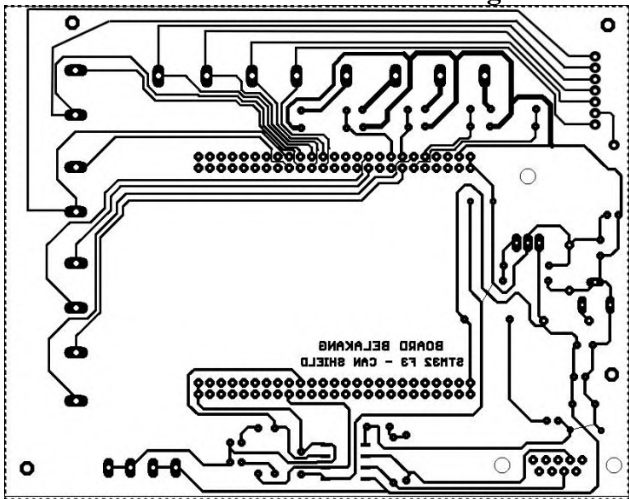
```
{
{return;}
}
```

LAMPIRAN 6 *Circuit Board Joulemeter (BMS)***Bagian Atas****Bagian Bawah**

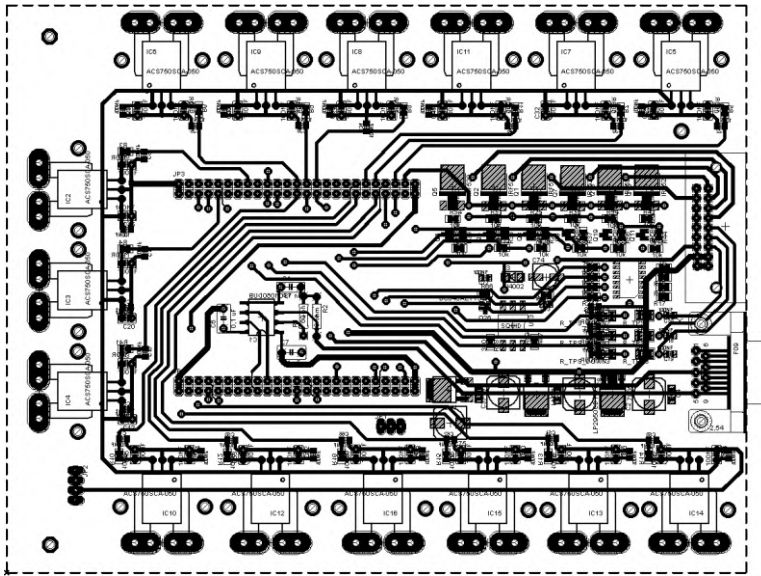
LAMPIRAN 7 *Circuit Board Xbee*



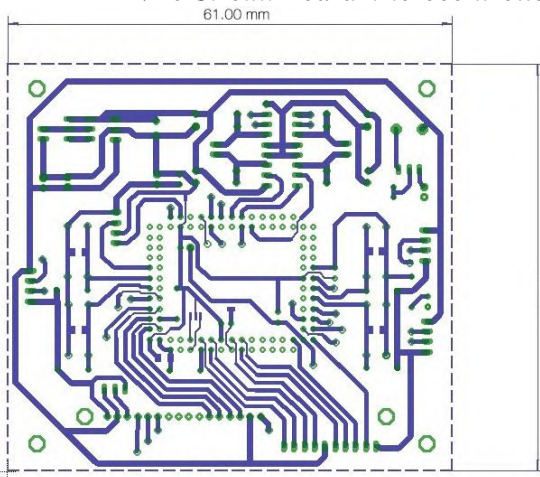
LAMPIRAN 8 *Circuit Board Xbee dengan Micro SD*



LAMPIRAN 9 *Circuit Board MPPT*



LAMPIRAN 10 *Circuit Board Microcontroller*



LAMPIRAN 11 Regulasi Pemakaian Gelombang Radio Untuk Amatir Australia

6 Use of an amateur station

The licensee:

- (a) must use an amateur station solely for the purpose of:
 - (i) self training in radiocommunications; or
 - (ii) intercommunications; or
 - (iii) technical investigations into radiocommunications; or
 - (iv) transmitting news and information services related to the operation of amateur stations, as a means of facilitating intercommunication; and
- (b) must not use an amateur station for financial gain; and
- (c) must not transmit:
 - (i) a message that is, or includes, an advertisement; or
 - (ii) any form of entertainment.

39A Devices operating in the frequency band 433.05–434.79 MHz

- (1) For paragraph 4 (1) (b) of the *Radiocommunications (Low Interference Potential Devices) Class Licence 2000*, a radiocommunications device operating in the frequency band 433.05–434.79 MHz in accordance with that Class Licence (except for that paragraph) is taken not to cause interference to an amateur repeater station.
- (2) The licensee accepts that subsection (1) applies in relation to interference to an amateur repeater station caused by a radiocommunications device mentioned in that subsection.

Indonesia

- Pada pita-pita frekuensi 156-156,7625 MHz, 156,8375-157,45 MHz, 160,6-160,975 MHz dan 161,475-162,05 MHz, tiap Administrasi memberi prioritas untuk servis bergerak maritim (Lihat Footnote RR 5.226). Keterangan rinci dapat dilihat pada Bab - 8 dokumen ini
- Pita Frekuensi 287 - 294 MHz dan 310 - 324 MHz di wilayah Jakarta dan sekitarnya telah dialokasikan untuk layanan *Broadband Wireless Access* (BWA) Multimedia. Frekuensi dimaksud bukanlah frekuensi *standard mass market* untuk BWA multimedia, TV digital, dsb, sehingga penyelenggara terkait harus mengembangkan produksinya sendiri. Keterangan rinci dapat dilihat pada bab - 9 dokumen ini
- Pita frekuensi 430 - 438 MHz digunakan bersama sekunder untuk amatir radio dan 435-438 MHz untuk amatir satelit.
- Pita frekuensi 438 - 450 MHz, 457.5 - 460 MHz, 467.5-470 MHz dialokasikan untuk kepentingan pertahanan. Pita frekuensi 450 - 457.5 dan 460 - 467,5 MHz dialokasikan untuk penyelenggara bergerak selular. Keterangan rinci bisa dilihat di Bab - 3 dokumen ini.

LAMPIRAN 12 Tabel Hasil Monitoring

1. S = *speed* (km/h)
2. TP = *throttle position* (%)
3. SP = *set point rpm*
4. PV = *pack voltage* (V)
5. P = *Panel* (A)
6. PC = *pack current* (A)
7. CCL = *Charge Current Limit*
8. DCL = *Discharge Current Limit*
9. SOC = *State of Charge*
10. AH = *ampere hour* (Ah)
11. 12v = *12 v supply* (v)
12. BHT = *batt highest temp* (°C)
13. BLT = *batt lowest temp* (°C)
14. BAT = *batt average temp* (°C)
15. BHC = *batt highest cell* (V)
16. BLC = *batt lowest cell* (V)
17. BAC = *batt average cell* (V)
18. BDV = *batt delta voltage* (V)

Jam	Data Hasil Monitoring								
	S	TP	SP	PV	P	PC	CCL	DCL	SOC
8:49:16	7	38,062	172,8	110		3,84	64	240	94

8:49:28	7	0,441	2	111		0,14	46	240	94
8:49:35	0	0	0	110		7,04	89	240	94
8:50:02	1	47,357	215	110		2,54	60	240	94
8:50:03	2	47,665	216,4	110		2,64	62	240	94
8:50:05	3	51,013	231,6	110		3,44	65	240	94
8:50:06	4	52,379	237,8	110		4,04	70	240	94
8:50:07	4	45,595	207	110		2,34	62	240	94
8:50:08	5	38,326	174	110		1,94	59	240	94
8:50:09	5	31,057	141	110		1,64	56	240	94
8:50:10	5	27,511	124,9	110		2,04	57	240	94
8:50:11	5	38,48	174,7	110		2,44	60	240	94
8:50:12	6	45,661	207,3	110		3,54	67	240	94
8:50:13	7	40,419	183,5	110		1,44	63	240	94
8:50:14	7	33,722	153,1	110		1,34	56	240	94
8:50:15	7	29,537	134,1	110		1,14	55	240	94
8:50:17	7	25,595	116,2	111		0,94	52	240	94
8:50:18	7	19,273	87,5	111		0,34	48	240	94

8:50:19	7	16,454	74,7	111		0,24	48	240	94
8:50:20	7	16,167	73,4	111		0,34	48	240	94
8:50:21	7	15,947	72,4	111		0,34	48	240	94
8:50:22	7	15,969	72,5	111		0,34	48	240	94
8:50:23	7	16,057	72,9	111		0,14	47	240	94
8:50:24	7	15,903	72,2	111		0,24	47	240	94
8:50:26	7	16,035	72,8	111		0,24	47	240	94
8:50:27	7	27,819	126,3	111		0,94	51	240	94
8:50:28	7	37,533	170,4	110		3,14	57	240	94
8:50:29	8	43,062	195,5	110		2,84	63	240	94
8:50:30	8	43,04	195,4	110		3,14	62	240	94
8:50:31	9	40,33	183,1	110		2,44	64	240	94
8:50:32	9	37,004	168	110		3,64	62	240	94
8:50:33	10	33,811	153,5	110		3,84	69	240	94
8:50:34	10	29,868	135,6	110		3,44	66	240	94
8:50:35	11	28,612	129,9	110		2,94	61	240	94
8:50:37	11	28,304	128,5	110		2,04	60	240	94

8:50:38	11	28,37	128,8	110		2,14	57	240	94
8:50:39	12	28,216	128,1	110		2,24	60	240	94
8:50:40	12	23,524	106,8	110		1,54	56	240	94
8:50:41	13	22,401	101,7	110		1,54	55	240	94
8:50:42	13	22,313	101,3	110		1,54	56	240	94
8:50:43	13	22,291	101,2	110		1,64	57	240	94
8:50:44	13	22,313	101,3	110		1,94	58	240	94
8:50:45	14	22,357	101,5	110		1,84	60	240	94
8:50:46	14	22,379	101,6	110		1,84	59	240	94
8:50:48	15	18,414	83,6	110		1,24	53	240	94
8:50:49	15	16,476	74,8	110		1,34	55	240	94
8:50:50	16	16,388	74,4	110		1,14	53	240	94
8:50:51	16	16,233	73,7	110		1,34	56	240	94
8:50:52	16	18,216	82,7	110		1,74	56	240	94
8:50:53	16	18,194	82,6	110		1,64	56	240	94
8:50:54	17	18,392	83,5	110		1,84	58	240	94
8:50:55	17	18,194	82,6	110		1,84	57	240	94

8:50:56	17	14,493	65,8	111		0,64	55	240	94
8:50:57	18	3,436	15,6	111		- 0,56	46	240	94
8:50:59	17	0,683	3,1	111		- 3,46	0	240	94
8:51:00	16	0	0	111		- 2,96	0	240	94

Jam	Data Hasil Monitoring								
	AH	12 v	BHT	BLT	BAT	BHC	BLC	BAC	BDV
8:49:16	113,9	11,6	35	32	33	4,16	4,03	4,1	0,123
8:49:28	113,9	11,6	35	32	33	4,17	4,1	4,12	0,067
8:49:35	113,9	11,5	35	32	33	4,14	3,96	4,08	0,179
8:50:02	113,9	11,6	35	32	33	4,16	4,05	4,1	0,105
8:50:03	113,8	11,6	35	32	33	4,16	4,06	4,1	0,104
8:50:05	113,9	11,6	35	32	33	4,16	4,03	4,1	0,12
8:50:06	113,9	11,6	35	32	33	4,15	4,02	4,1	0,129
8:50:07	113,9	11,6	35	32	33	4,16	4,06	4,1	0,099

8:50:08	113,9	11,6	35	32	33	4,16	4,07	4,11	0,09
8:50:09	113,9	11,6	35	32	33	4,16	4,08	4,11	0,083
8:50:10	113,9	11,6	35	32	33	4,16	4,08	4,11	0,083
8:50:11	113,9	11,6	35	32	33	4,16	4,06	4,1	0,102
8:50:12	113,9	11,6	35	32	33	4,16	4,03	4,1	0,126
8:50:13	113,8	11,6	35	32	33	4,16	4,08	4,11	0,081
8:50:14	113,9	11,6	35	32	33	4,17	4,09	4,11	0,08
8:50:15	113,9	11,6	35	32	33	4,17	4,09	4,11	0,074
8:50:17	113,9	11,6	35	32	33	4,17	4,1	4,11	0,071
8:50:18	113,9	11,6	35	32	33	4,17	4,1	4,11	0,066
8:50:19	113,9	11,6	35	32	33	4,17	4,1	4,11	0,067
8:50:20	113,9	11,6	35	32	33	4,17	4,1	4,11	0,066
8:50:21	113,9	11,6	35	32	33	4,17	4,1	4,11	0,067
8:50:22	113,9	11,6	35	32	33	4,17	4,1	4,11	0,066
8:50:23	113,9	11,6	35	32	33	4,17	4,1	4,12	0,067
8:50:24	113,9	11,6	35	32	33	4,17	4,1	4,11	0,066
8:50:26	113,9	11,6	35	32	33	4,17	4,1	4,11	0,067

8:50:27	113,9	11,6	35	32	33	4,17	4,09	4,11	0,072
8:50:28	113,9	11,6	35	32	33	4,16	4,05	4,1	0,111
8:50:29	113,9	11,6	35	32	33	4,16	4,05	4,1	0,11
8:50:30	113,9	11,6	35	32	33	4,16	4,04	4,1	0,114
8:50:31	113,9	11,6	35	32	33	4,16	4,06	4,1	0,096
8:50:32	113,9	11,6	35	32	33	4,16	4,03	4,1	0,128
8:50:33	113,8	11,6	35	32	33	4,16	4,03	4,1	0,128
8:50:34	113,9	11,6	35	32	33	4,16	4,04	4,1	0,117
8:50:35	113,9	11,6	35	32	33	4,16	4,06	4,1	0,096
8:50:37	113,9	11,6	35	32	33	4,16	4,07	4,11	0,092
8:50:38	113,9	11,6	35	32	33	4,16	4,08	4,11	0,086
8:50:39	113,9	11,6	35	32	33	4,16	4,07	4,1	0,093
8:50:40	113,9	11,6	35	32	33	4,16	4,08	4,11	0,081
8:50:41	113,9	11,6	35	32	33	4,16	4,08	4,11	0,08
8:50:42	113,9	11,6	35	32	33	4,16	4,08	4,11	0,086
8:50:43	113,9	11,6	35	32	33	4,16	4,08	4,11	0,083
8:50:44	113,9	11,6	35	32	33	4,16	4,07	4,11	0,089

8:50:45	113,8	11,6	35	32	33	4,16	4,08	4,11	0,084
8:50:46	113,9	11,6	35	32	33	4,16	4,08	4,11	0,081
8:50:48	113,9	11,6	35	32	33	4,17	4,09	4,11	0,078
8:50:49	113,9	11,6	35	32	33	4,17	4,09	4,11	0,077
8:50:50	113,9	11,6	35	32	33	4,17	4,09	4,11	0,072
8:50:51	113,9	11,6	35	32	33	4,16	4,08	4,11	0,081
8:50:52	113,9	11,6	35	32	33	4,16	4,08	4,11	0,081
8:50:53	113,9	11,6	35	32	33	4,16	4,08	4,11	0,086
8:50:54	113,9	11,6	35	32	33	4,16	4,07	4,11	0,086
8:50:55	113,9	11,6	34	32	33	4,16	4,08	4,11	0,086
8:50:56	113,9	11,6	34	32	33	4,17	4,1	4,11	0,067
8:50:57	113,9	11,6	34	32	33	4,17	4,11	4,12	0,066
8:50:59	113,9	11,6	35	32	33	4,2	4,12	4,14	0,077
8:51:00	113,9	11,6	34	32	33	4,2	4,12	4,13	0,082

LAMPIRAN 13 *Bill of Material* dan *Bill of Quantity* Sistem Monitoring Widya Wahana V

NO	NAMA ITEM	UNIT	QUANTITY	HARGA (Rupiah)		KEPUTUSAN
				SATUAN	TOTAL	
A.	Sistem Pengambilan Data					
1.	Kabel Connector CAN	Buah	2	Rp200.000	Rp400.000	Beli
2.	Kabel	Meter	2	Rp5.000	Rp10.000	Beli
3.	Sensor Arus (LEM DHAB)	Buah	1	Rp743.208	Rp743.208	Beli
4.	Board Joulemeter					
4.1	Mikrokontroller STM32F3	Buah	1	Rp250.000	Rp250.000	Beli
4.2.	Board	M ²	1	Rp100.000	Rp100.000	Beli
4.3.	Connector 2x25	Buah	2	Rp5.000	Rp10.000	Beli
4.4.	Resistor 50k	Buah	15	Rp10.000	Rp150.000	Beli
4.5.	Resistor 1k	Buah	15	Rp600	Rp9.000	Beli
4.6.	Resistor 330	Buah	1	Rp300	Rp300	Beli
4.7.	Resistor 220	Buah	5	Rp200	Rp1.000	Beli

4.8.	Resistor 4k7	Buah	12	Rp600	Rp7.200	Beli
4.9.	Polarized Capacitor(C1000uf)	Buah	2	Rp1.500	Rp3.000	Beli
4.10.	Capacitor (C100nf)	Buah	2	Rp700	Rp1.400	Beli
4.11.	Voltage Regulator(IC7805T)	Buah	1	Rp3.000	Rp3.000	Beli
4.12.	LED 3mm	Buah	1	Rp500	Rp500	Beli
4.13.	Connector Header(AK300/12)	Buah	1	Rp4.000	Rp4.000	Beli
4.14.	Connector Header(AK300/4)	Buah	7	Rp1.000	Rp7.000	Beli
4.15.	Connector Header(AK300/6)	Buah	1	Rp2.000	Rp2.000	Beli
4.16.	Connector Header(AK300/2)	Buah	2	Rp500	Rp1.000	Beli
4.17.	Tiny Switch(TL39WO)	Buah	2	Rp15.000	Rp30.000	Beli
4.18.	Dioda (1N4004)	Buah	2	Rp250	Rp500	Beli
4.19.	Pin Header(MA04-1)	Buah	2	Rp1.000	Rp2.000	Beli

4.20.	Isolated CAN Tranceiver (ISO1050DUB)	Buah	1	Rp73.582	Rp73.582	Beli
4.21.	Push Switch	Buah	2	Rp5.000	Rp10.000	Beli
4.22.	Sharp Optocoupler (PC817)	Buah	6	Rp3.000	Rp18.000	Beli
4.23.	LED Opto	Buah	6	Rp500	Rp3.000	Beli
4.24.	N-Channel Power MOSFET (IRF540)	Buah	6	Rp5.000	Rp30.000	Beli
B.	Sistem Penampilan Data LCD					
1.	Kabel Connector Paralel	Buah	1	Rp50.000	Rp50.000	Beli
2.	Mikrokontroller Arduino UNO	Buah	1	Rp185.000	Rp185.000	Beli
3.	LCD TFT 4"	Buah	1	Rp200.000	Rp200.000	Beli
C.	Sistem Pengiriman Data					
1.	Antena Omni	Buah	1	Rp55.000	Rp55.000	Beli
2.	Modul Xbee	Buah	1	Rp275.000	Rp275.000	Beli

3.	Kabel Connector Serial	Buah	1	Rp50.000	Rp50.000	Beli
4.	Board Xbee					
4.1	Wago Screw Clamp (W239-104)	Buah	3	Rp10.000	Rp30.000	Beli
4.2.	Wago Screw Clamp (W237-4)	Buah	1	Rp7.500	Rp7.500	Beli
4.3.	Wago Screw Clamp (W239-302)	Buah	2	Rp11.000	Rp22.000	Beli
4.4.	Polarized Capacitor(C1000uf)	Buah	2	Rp1.500	Rp3.000	Beli
4.5.	Capacitor (22pf)	Buah	2	Rp300	Rp600	Beli
4.6.	Voltage Regulator (IC317T)	Buah	1	Rp2.100	Rp2.100	Beli
4.7.	Resistor 330	Buah	3	Rp300	Rp900	Beli
4.8.	Z Dioda (1N4728)	Buah	1	Rp250	Rp250	Beli
4.9.	Dioda (1N4004)	Buah	1	Rp250	Rp250	Beli
4.10.	Power Jack 2.5mm	Buah	1	Rp1.000	Rp1.000	Beli
4.11.	LED	Buah	3	Rp500	Rp1.500	Beli
4.12.	SUB-D F094	Buah	1	Rp50.000	Rp50.000	Beli

4.13.	Connector 2x25	Buah	2	Rp5.000	Rp10.000	Beli
4.14.	Capacitor 4.7nf	Buah	1	Rp1.000	Rp1.000	Beli
4.15.	Resistor 60	Buah	2	Rp200	Rp400	Beli
4.16.	Capacitor 0.1uf	Buah	2	Rp200	Rp400	Beli
4.17.	Isolated CAN Tranceiver (ISO1050DUB)	Buah	1	Rp73.582	Rp73.582	Beli
4.18.	Resistor 220	Buah	2	Rp200	Rp400	Beli
4.19.	Board	M ²	1	Rp100.000	Rp100.000	Beli
4.20.	Mikrokontroller STM32F3	Buah	1	Rp250.000	Rp250.000	Beli
D.	Sistem Penerimaan Data					
1.	Antena Corner Reflector					
1.1	Siku Aluminium	Meter	14	Rp3.500	Rp49.000	Beli
1.2	Keling	Buah	84	Rp100	Rp8.400	Beli
1.3	Baut 6mm	Buah	16	Rp150	Rp2.400	Beli
1.4	Mur 6mm	Buah	16	Rp150	Rp2.400	Beli

“Halaman ini sengaja dikosongkan”

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah *review* dan analisa dari sistem monitoring Widya Wahana V, maka didapatkan kesimpulan sebagai berikut:

1. Sistem monitoring pada mobil Widya Wahana V menggunakan komponen-komponen sebagai berikut TFT 4" LCD, 3 mikrokontroller, antena omni, panel surya, MPPT, BMS, baterai, kontroller motor, dan motor listrik. Data yang diambil berupa rpm, kondisi baterai, dan arus. Data akan diolah pada joulemeter. Data tersebut akan ditampilkan pada layar LCD dan data juga dikirimkan ke mobil *escort* melalui antena. Sistem monitoring pada mobil *escort* menggunakan komponen-komponen sebagai berikut antena *corner reflector*, laptop, modul 3DR, dan *booster* antena. Data dari mobil Widya Wahana V akan diterima melalui antena *corner reflector*. Data yang telah diterima akan ditampilkan pada display laptop. *Software* yang digunakan untuk membuat sistem monitoring yaitu dengan *software* QT.
2. Data hasil *real-time* sistem monitoring yang didapat yaitu kecepatan, *throttle position*, *set point rpm*, *pack voltage*, arus panel surya, *pack current*, *charge current limit*, *discharge current limit*, *state of charge*, *ampere hour*, *12 volt supply*, *batt highest temp*, *batt lowest temp*, *batt average temp*, *batt highest cell*, *batt lowest cell*, *batt average*, dan *batt delta voltage*.
3. Analisa sistem monitoring ini bertujuan untuk melihat kekurangan-kekurangan yang terjadi pada sistem monitoring. Sebagai berikut analisa sistem monitoring:
 - Dimensi *discovery board* STM32F3 masih terlalu besar.
 - Sistem pembacaan arus
 - Sistem penyimpanan data

- Kinerja baterai tiap sel

5.2 Saran

Berdasarkan *review* dan analisa sistem monitoring Widya Wahana V dalam Tugas Akhir ini terdapat beberapa saran untuk menunjang pengembangan sistem monitoring pada mobil Widya Wahana VI, antara lain :

1. *Discovery board* STM32F3 ini diganti dengan membuat *board* sendiri yang lebih kecil dan sederhana dengan memasang mikro STM32F3 sesuai dengan kebutuhan dalam sistem.
2. Kabel *output* tiap MPPT diberi 1 sensor arus (LEM DHAB) dan dibuatkan *board* tersendiri untuk mengambil dan mengolah data arus dengan begitu arus yang keluar tiap MPPT dapat terbaca atau tertampil.
3. *Board* XBEE pada mobil Widya Wahana V diberi slot untuk *micro SD* atau *SD card* dan pada *board* XBEE diberi tambahan program untuk penyimpanan data pada *micro SD*.
4. Pengambilan data pada BMS dibuatkan *board* sendiri yang tidak tergabung dengan pengambilan data yang lain dan pada *board* tersebut diberi mikro STM32F3.

DAFTAR PUSTAKA

- 3DRobotics. 2013. 3DR Radio Module.
<https://3drobotics.com/wp-content/uploads/2013/10/3DR-Radio-V2-doc1.pdf>.
8 November 2015.
- Akbar, Ali., dkk. 2015. *Radio Modul FTETI 1 Sistem Telemetri*. dalam jurnal Program Studi Fisika Institut Teknologi Bandung. Bandung: Institut Teknologi Bandung
- Antenna. Antenna Basics. <http://wireless.ictp.it/handbook/C4.pdf>.
8 November 2015.
- Antenna. Antenna Types.
<https://www.ncjrs.gov/pdffiles1/nij/185030b.pdf>. 8 November 2015.
- Chang, Heather., dkk. 2010. *Wireless Telemetry System for Solar Car*. ECE4007 Senior Design Project, Section L01.
- Digi International. 2008. Xbee-PRO Module.
<https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-900-Datasheet.pdf>. 8 November 2015.
- Elisa. 2010. Sistem Telemetri.
[elisa.ugm.ac.id/user/archive/download/50230/9a170dc5b3b898efd406f4bfd763f730 sistem telemetri](http://elisa.ugm.ac.id/user/archive/download/50230/9a170dc5b3b898efd406f4bfd763f730_sistem_telemetri).
8 November 2015.
- Japan Automobile Manufacturers Association. 2004. *Guidelines for In-vehicle Display Systems*. Version 3.0.
Jepang.
- Komarudin, M., dkk. 2012. *Sistem Telemetri Unmanned Aerial Vehicle (UAV) Berbasis Inertial Measurement Unit (UMI)*. dalam Elektrika Jurnal Rekayasa dan Teknologi Elektro, Volume 6, No. 3. Bandar Lampung: Universitas Lampung
- LEM. 2009. LEM DHAB S/34. http://www.hg-electronics.de/downloads/LEM/LEM_Katalog_Industrie.pdf. 8 November 2015.

- Munarso, dkk. 2014. *Sistem Telemetry Pemantauan Suhu Lingkungan Menggunakan Mikrokontroler Dan Jaringan WIFI*. Youngster Physics Journal, Vol. 3, No. 3, 249-256. Semarang: Universitas Diponegoro.
- Nisa, Siti Choirun. 2014. *Pemodelan dan Simulasi Mobil Sapu Angin Surya Generasi Kedua Guna Memaksimalkan Performa Di World Solar Challenge 2015*. Tugas Akhir pada Institut Teknologi Sepuluh Nopember. Surabaya: Institut Teknologi Sepuluh Nopember.
- Orion BMS. 2014. Battery Management System. <http://www.orionbms.com/>. 20 Maret 2014.
- SOLBIAN. 2014. SOLBIAN Flex sp137. <http://www.selfenergy.pl/images/pdf/SP137.pdf>. 8 November 2015.
- Sotyaramadhani, Grangsan. 2014. *Perancangan Integrater Real-Time Monitoring System untuk Mobil Listrik Ezzy ITS*. Tugas Akhir pada Institut Teknologi Sepuluh Nopember. Surabaya: Institut Teknologi Sepuluh Nopember.
- UA Solar Racing Team. 2010. *Solar Car Power Monitoring System Final Technical Report*. Interdisciplinary Engineering Design Program University of Arizona. Arizona: Universitas Arizona.
- Weiss, Betina, dkk. 2007. *Introduction to Microcontroller*. Version 1.4. Vienna: Vienna University of Technology Institute of Computer Engineering.
- Wikipedia. 2015. Voltage Divider. https://en.wikipedia.org/wiki/Voltage_divider. 8 November 2015.

BIODATA PENULIS



Hisyam Fikriy dilahirkan di Surabaya, 12 Juni 1993. Anak pertama dari Chairul Imron dan Tutik Indarni. Penulis menyelesaikan masa studi di SD Luqman Al-Hakim Surabaya pada tahun 2004, dilanjutkan ke SMP Negeri 29 Surabaya pada tahun 2009, dan SMA Negeri 3 Surabaya pada tahun 2011.

Selepas SMA penulis melanjutkan studinya di Institut Teknologi Sepuluh Nopember Jurusan Teknik Mesin pada tahun ajaran 2011/2012. Selama kuliah di Teknik

Mesin ITS penulis aktif mengikuti organisasi Lembaga Bengkel Mahasiswa Mesin. Tidak hanya itu penulis juga telah bergabung dalam *ITS Solar Car Racing Team* yang telah berkompetisi di Australia pada *World Solar Challenge 2015*. Di Teknik Mesin ITS penulis memilih untuk masuk Laboratorium Otomasi dan mengerjakan tugas akhir dengan topik sistem monitoring dibawah bimbingan Dr.Ir.Muhammad Nur Yuniarto. Pada tahun 2016 penulis menyelesaikan studi S1-nya.

“Halaman ini sengaja dikosongkan”